AD-A048 481 REGIS COLL WESTON MASS F/G 4/2

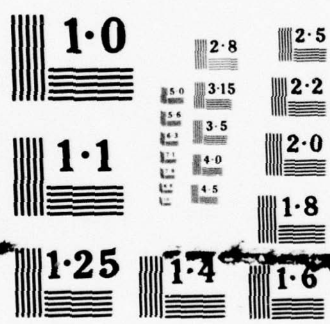
A GENERALIZED COMPUTER PROGRAM FOR PRIMITIVE-EQUATION MODELS.(U)

MAY 77 T J LEONARD, J C METTAUER F19628-77-C-0010

UNCLASSIFIED SCIENTIFIC-1 AFGL-TR-77-0183 NL

1 OF 2

AD A048481



AD A048481

AFGL-TR-77-0183

A GENERALIZED COMPUTER PROGRAM FOR PRIMITIVE-EQUATION MODELS

Thomas J. Leonard
Jack C. Mettauer

Regis College Research Center
235 Wellesley Street
Weston, Massachusetts 02193

Scientific Report No. 1

30 May 1977

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 18 AFGL-TR-77-0183	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: 9 Rept. for	
4. TITLE (and Subtitle): 6 A Generalized Computer Program for Primitive-Equation Models.	5. TYPE OF REPORT & PERIOD COVERED: Scientific Report # 1, 1 October 1976-30 May 1977
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): 10 Thomas J. Leonard, Jack C. Mettauer	8. CONTRACT OR GRANT NUMBER(s): new 15 F19628-77-C-0010
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Regis College, 235 Wellesley Street, Weston, Massachusetts 02193	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62101F, 16 2310G2AF 17 G2
11. CONTROLLING OFFICE NAME AND ADDRESS: 14 SCIENTIFIC-1	12. REPORT DATE: 11 30 May 1977
	13. NUMBER OF PAGES: 144
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Air Force Geophysics Laboratories, Hanscom AFB, Massachusetts 01731, Contract Monitor: Peter Giorgio LY 12 146p.	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

GLOBAL	time integration scheme
"box method"	contour level
primitive equations	file
prognostic equations	buffer
diagnostic equations	space operator

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This program is an attempt at a large scale, multi-level general circulation model of the atmosphere. The original conception was to implement a very flexible program. The major programs are: INITE (Since a typical problem will require more storage than can be accomodated, this program parcels the data and codes information as to how it is partitioned.); GEX (A solution program to step the fields ahead in time.); and DISPLY (This program displays the various fields which have been saved from the GEX run.).

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

301000

TABLE OF CONTENTS

I. Purpose of Program

This program is an attempt at a large scale, multi-level general circulation model of the atmosphere. The original conception was to implement a very flexible program, not committed to any fixed spherical grid, to any fixed difference scheme, or any number of prognostic variables or form for the equations (although, at this point in the evolution of the program, the set of primitive equations programmed, were those found in Kurihara and Holloway (1967)).

Global is programmed for the $(GS)^2$ grid system, which accomodates a wide variety of spherical grids including the Kurihara grid. The solution program RHS employs the "box method," as developed in Kurihara and Holloway.

Since a typical problem will require more storage than can be accomodated in core, it is necessary to parcel the data, and so, to code information as to how it is partitioned. This is done in a preliminary program INITE. Then a solution program, GEX, steps the fields ahead in time. For efficient operation only that data, which is necessary for updating one latitude band with no computational redundancy, is accomodated in core. Finally, a display program, DISPLY, displays the various fields which have been saved from the GEX run.

II. Literature

In order to understand the operation of this program, it is necessary to have a familiarity with at least

(i) Kurihara Yoshio: "Numerical Integration of The Primitive Equations on a Spherical Grid," Monthly Weather Review: Vol. 93, No.1, July 1965.

(ii) Kurihara, Yoshio and Holloway, J. Leith: "Numerical Integration of a nine level Global Primitive Equations Model Formulated by the Box Method," Monthly Weather Review: Vol. 95, No. 8, August 1967.

(iii) Klein, R. and Mettauer, J., "The Numerical Solution of Partial Differential Equations," January 1969.

(iv) Klein, Mettauer, Maglione, Spiegel: "Certain Finite Difference Methods for the Solution of Large Scale Circulation Problems," Final Report, January 1, 1972.

(v) Klein <u>et al.</u>: "Certain Finite Difference Methods for the Solution of Large Scale Circulation Problems II," November 8, 1973.

(vi) Ahlberg, Nilson, Wash: "The Theory of Splines and Their Applications," Academic Press, 1967.

We shall hereafter refer to these papers as

(i) K paper

(ii) KH paper

(iii) KM paper

(iv) KMI paper

(v) KMII paper

(vi) ANW paper.

III. Description of Model

For the sake of completeness, the model equations below are taken from the KH paper.

The equations of motion in λ, θ, σ coordinates are:

$$\frac{\partial(P_* u)}{\partial t} = -D_3(u) + \left(f + \frac{\tan\theta}{a} u\right) P_* v$$

$$- R \frac{\partial P_* T}{\alpha \partial \lambda} - \frac{\partial}{\partial \sigma}\left(P_* \sigma \frac{\partial \phi}{\alpha \partial \lambda}\right) + F_\lambda$$

$$\frac{\partial(P_* v)}{\partial t} = -D_3(v) - \left(f + \frac{\tan\theta}{a}\ u\right) P_* u$$

$$- R\ \frac{\partial P_* T}{a\partial\theta} - \frac{\partial}{\partial\sigma}\left(P_*\sigma\frac{\partial\phi}{a\partial\theta}\right) + F_\theta$$

where f is the Corolis parameter; R, the gas constant for dry air; T, temperature; ϕ, geopotential of a sigma surface; F_λ and F_θ, frictional forces in the zonal and meridional directions respectively; and D_3 the three-dimensional divergence operator defined in KH. The first law of thermodynamics is

$$\frac{\partial(P_* T)}{\partial t} = -D_3(T) + \frac{R}{c_p}\ \frac{T\omega}{\sigma} + \frac{P_*\dot{q}}{c_p} + F_T$$

where c_p is the specific heat capacity at constant pressure; ω vertical P-velocity $\frac{dP}{dt}$; $\dot{q}$ added heat per unit mass; and F_T, the effect of thermal diffusion. The continuity equation takes the form:

$$\frac{\partial P*}{\partial t} = -\int_0^1 D_2(1)\ d\sigma$$

where D_2 is the two dimensional divergence operator defined in KH.

The vertical σ and P-velocities are obtained by the diagnostic relations:

$$\bar{\omega} = \frac{1}{P_*}\left[\sigma\int_0^1 D_2(1)\ d\sigma - \int_0^\sigma D_2(1)\ d\sigma\right]$$

$$\omega = P_*\ \bar{\omega} + \sigma\left[\frac{\partial P_*}{\partial t} + u\ \frac{\partial P_*}{\alpha\partial\lambda} + v\ \frac{\partial P_*}{a\partial\theta}\right]$$

Finally, the hydrostatic relation is:

$$\phi - RT = \frac{\partial\phi\sigma}{\partial\sigma}$$

GLOBAL integrates, in time, finite difference versions of these equations which are defined on a generalized grid (GS)[2]. In deriving the finite difference forms of the original differential equations, it is desirable to preserve conservation properties. This can be accomplished by employing the "box method" as an overall strategy. That is, if

$$\frac{\partial x}{\partial t} = -\nabla\cdot(\vec{v}x) + F$$

where x is a scalar quantity; $\vec{v}$, a three dimensional velocity vector; $\nabla\cdot$, a three dimensional divergence operator; and F a source term of the quantity x; then integrating around a volume element or a box, and employing the Gauss Theorem gives:

$$\frac{\partial \overline{x}}{\partial t} = -\frac{1}{V}\int_S \nu_n \, x \, ds + \frac{1}{V}\int F \, dV$$

where $\overline{x}$ is a mean value of x for the box with volume V and surface S; and ν_n is the outward component of velocity on the box surface.

Then, the finite difference forms of the prognostic equations become:

$$(1) \qquad \frac{\partial(P_{*o}u_o)}{\partial t} = -D\left[\frac{u_l+u_o}{2}\right] + (\hat{f} + \hat{m}u_o)\, P_{*o}v_o$$

$$- RG_\lambda(P_*T) - \frac{\delta_k[\overline{\sigma}\cdot L_\lambda(P_*,\phi)]}{\delta_k\sigma} + (F_\lambda)_o$$

$$(2) \qquad \frac{\partial(P_{*o}v_o)}{\partial t} = -D\left[\frac{v_l+v_o}{2}\right] - (\hat{f} + \hat{m}u_o)\, P_{*o}v_o$$

$$- RG_\theta(P_*T) - \frac{\delta_k[\overline{\sigma}\cdot L_\theta(P_*,\phi)]}{\delta_k\sigma} + (F_\theta)_o$$

(3) $$\frac{\partial(P_{*o}T_o)}{\partial t} = -D\left(\frac{T_1 + T_o}{2}\right) + \frac{R}{c_p}\,\frac{T_o\omega_o}{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})} + \frac{P_{*o}}{c_p}\hat{\dot{q}} + (F_T)_o$$

(4) $$\frac{\partial(P_{*o})}{\partial t} = -\sum_{k=1}^{k} H_1(1)\,\Delta_k\sigma$$

where the o subscript indicates an average value of the variable; taken to be the value of the center of the box, and where

$$\hat{f} = \frac{1}{\Delta V}\iiint\limits_{\Delta V} f\,dV$$

$$\hat{m} = \frac{1}{\Delta V}\iiint\limits_{\Delta V} \frac{\tan\theta}{a}\,dV$$

$$\hat{\dot{q}} = \frac{1}{\Delta V}\iiint\limits_{\Delta V} \dot{q}\,dV$$

and the space operators D, H, G_λ, G_θ, L_λ, L_θ are as defined in KH.

The diagnostic relations are:

(5) $$\overline{\omega}_{k+\frac{1}{2}} = \frac{1}{P_{*o}}\left[\sigma_{k+\frac{1}{2}}\sum_{k=1}^{k}\{H_1(1)\,\Delta_{k'}\sigma\} - \sum_{k'=1}^{k}\{H_1(1)\,\Delta_{k'}\sigma\}\right]$$

(6) $$\omega_{ok} = P_{*\sigma}\frac{\overline{\omega}_{k+\frac{1}{2}} + \overline{\omega}_{k-\frac{1}{2}}}{2} + \frac{\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}}}{2}\left[\frac{\partial P_{*o}}{\partial t} + u_o G_\lambda(P_*) + v_o G_\theta(P_*)\right]$$

and

$$\phi_k - RT_k = \frac{\delta_k(\phi\sigma)}{\delta_k\sigma}$$

which leads to the expression for ϕ at the top and bottom of a box:

(7) $$\phi_{k-\frac{1}{2}} = \phi_{k+\frac{1}{2}} + \Delta_k\sigma \frac{RT_k}{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})}$$

The direction of calculation for each time step is:

1. Determine ϕ recursively at all levels from the diagnostic relation (7) using complete knowledge of $\phi_{k+\frac{1}{2}} = \phi_*$.
2. Calculate H_1 and $G_\lambda(P_*)$, $G_\theta(P_*)$.
3. Determine $\overline{\omega}$ at all levels from (5) where the boundary conditions are fixed $\overline{\omega}_{\frac{1}{2}} = \overline{\omega}_{k+\frac{1}{2}} = 0$.
4. Calculate the P_* tendency from (4).
5. Calculate ω at all levels from (6).
6. Calculate the space operators at all values of the arguments that appear in (1) thru (3).
7. Calculate the tendencies of the 3 dimensional prognostic variables from (1) thru (3).
8. Step the fields ahead using one of the time integration schemes.
9. Repeat the procedure the desired number of time steps.

IV. History

Check out on GLOBAL began with a series of baratropic runs. The first significant result was a leapfrog run on a coarse 16 x 32 grid and wave number one; which went out to about 60 days and then blew up. When the same run was made with wave number four the blow up occurred at 14 days (Δt was 360s in this run). The results were muddy at 7 days, and, in fact, there was a noticiable tilting in the v field after one day. It seemed that trouble was coming down from the poles, so it was decided to refine the grid. A 32 x 64 leapfrog, wave number 4 run, with Δt = 75s, ran out to 14 days without trouble. Total energy changed only in the 9^{th} decimal place.

Another approach to the problem was to mix Adams Bashford steps in with the leapfrog steps, in the hopes of stabilizing the run. If the mixing ratio was 2/100, the run went to 14 days. However, when the ratio was decreased to 1/500, the run blew up after 3327 time steps (3360 ts. ≡ 14 days).

To remedy the disruptive influence of small scale effects near the poles, various filters were tried. First, a program was written to do a spectral filter on the prognostic fields after every N time steps (N specified at object time). The program allowed the user to specify which latitude bands he wished filtered, and which wave numbers he wished to eliminate. When this filter was used to eliminate the 2 grid interval wave from the height field, and from the winds u, v, the plots showed sharp division at the transition between unfiltered bands (midlatitudes), and filtered bands (near the poles). To avoid the disadvantage of such a sharp truncation of wave number, a different filtering technique was applied both east-west and north-south on a 16 x 32 grid. The filter described in section V was applied with $p = 3$ to the coarse grid after every 10 time steps. The run resulted in fields with little amplitude, and left unchanged the pronounced tilting apparent in the unfiltered run. It appears that most of the smoothing is due to the difference scheme, and that this fact was disguised in the unfiltered run by the blow up.

Another approach, that was partially developed, was to replace the fields near the poles by weighted least squares polynomials, tying the polynomials not only to grid values, but to numerical derivatives near the poles. Replacing finite difference forms for space derivatives near the poles, with the polynomials' derivatives, and transforming back to the λ, θ coordinates, it would be possible to prognosticate avoiding the singularities that occur

in the finite difference forms when using either a polar box or a latitude-longitude grid.

If the projective space is defined by:

$$x = x(\lambda,\theta) = \cos\theta \cos\lambda$$

$$y = y(\lambda,\theta) = \cos\theta \sin\lambda$$

then

$$P(\lambda,\theta) \longrightarrow P(x,y)$$

and if we define:

$$J \equiv \begin{bmatrix} \frac{\partial x}{\partial \lambda} & \frac{\partial y}{\partial \lambda} \\ \frac{\partial x}{\partial \theta} & \frac{\partial y}{\partial \theta} \end{bmatrix} \qquad J^{-1} = \begin{bmatrix} DX \\ \hline DY \end{bmatrix}$$

where

$$DX = \begin{bmatrix} \frac{\partial \lambda}{\partial x} & \frac{\partial \theta}{\partial x} \end{bmatrix} \qquad DY = \begin{bmatrix} \frac{\partial \lambda}{\partial y} & \frac{\partial \theta}{\partial y} \end{bmatrix}$$

with

$$T \equiv J\,T \qquad U \equiv K\,T$$

and

$$K \equiv \begin{bmatrix} \left(\frac{\partial x}{\partial \lambda}\right)^2 & \frac{\partial^2 x}{\partial \lambda^2} & 2\,\frac{\partial x}{\partial \lambda}\frac{\partial y}{\partial \lambda} & \frac{\partial^2 y}{\partial \lambda^2} & \left(\frac{\partial y}{\partial \lambda}\right)^2 \\ 0 & \frac{\partial x}{\partial \lambda} & 0 & \frac{\partial y}{\partial \lambda} & 0 \\ \frac{\partial x}{\partial \lambda}\frac{\partial x}{\partial \theta} & \frac{\partial^2 x}{\partial \lambda \partial \theta} & \left[\frac{\partial x}{\partial \theta}\frac{\partial y}{\partial \lambda} + \frac{\partial x}{\partial \lambda}\frac{\partial y}{\partial \theta}\right] & \frac{\partial^2 y}{\partial \lambda \partial \theta} & \frac{\partial y}{\partial \lambda}\frac{\partial y}{\partial \theta} \\ 0 & \frac{\partial x}{\partial \theta} & 0 & \frac{\partial y}{\partial \theta} & 0 \\ \left(\frac{\partial x}{\partial \theta}\right)^2 & \frac{\partial^2 x}{\partial \theta^2} & 2\,\frac{\partial x}{\partial \theta}\frac{\partial y}{\partial \theta} & \frac{\partial^2 y}{\partial \theta^2} & \left(\frac{\partial y}{\partial \theta}\right)^2 \end{bmatrix}$$

and

$$dx \equiv \left[\left(\frac{\partial\lambda}{\partial x}\right)^2 \quad \frac{\partial^2\lambda}{\partial x^2} \quad 2\,\frac{\partial\lambda}{\partial x}\frac{\partial\theta}{\partial x} \quad \frac{\partial^2\theta}{\partial x^2} \quad \left(\frac{\partial\theta}{\partial x}\right)^2 \right]$$

$$dxy \equiv \left[\frac{\partial\lambda}{\partial x}\frac{\partial\lambda}{\partial y} \quad \frac{\partial^2\lambda}{\partial x\partial y} \quad \left[\frac{\partial\lambda}{\partial y}\frac{\partial\theta}{\partial x} + \frac{\partial\lambda}{\partial x}\frac{\partial\theta}{\partial y}\right] \quad \frac{\partial^2\theta}{\partial x\partial y} \quad \frac{\partial\theta}{\partial x}\frac{\partial\theta}{\partial y} \right]$$

$$dy \equiv \left[\left(\frac{\partial\lambda}{\partial y}\right)^2 \quad \frac{\partial^2\lambda}{\partial y^2} \quad 2\,\frac{\partial\lambda}{\partial y}\frac{\partial\theta}{\partial y} \quad \frac{\partial^2\theta}{\partial y^2} \quad \left(\frac{\partial\theta}{\partial y}\right)^2 \right]$$

so that

$$K^{-1} = \left[\begin{array}{ccccc} \multicolumn{5}{c}{dx} \\ \hdashline o & \frac{\partial\lambda}{\partial x} & o & \frac{\partial\theta}{\partial x} & o \\ \hdashline \multicolumn{5}{c}{dxy} \\ \hdashline o & \frac{\partial\lambda}{\partial y} & o & \frac{\partial\theta}{\partial y} & o \\ \hdashline \multicolumn{5}{c}{dy} \end{array}\right]$$

then the first derivatives transform as:

$$\begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \xrightarrow{\;J\;} \begin{bmatrix} \frac{\partial p}{\partial \lambda} \\ \frac{\partial p}{\partial \theta} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\frac{\partial \lambda}{\partial \lambda}} \\ \dot{\frac{\partial \lambda}{\partial \theta}} \\ \dot{\frac{\partial \theta}{\partial \lambda}} \\ \dot{\frac{\partial \theta}{\partial \theta}} \end{bmatrix} \xrightarrow{\begin{bmatrix} J^{-1} & 0 \\ 0 & J^{-1} \end{bmatrix}} \begin{bmatrix} \dot{\frac{\partial \lambda}{\partial x}} \\ \dot{\frac{\partial \lambda}{\partial y}} \\ \dot{\frac{\partial \theta}{\partial x}} \\ \dot{\frac{\partial \theta}{\partial y}} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \end{bmatrix} \xrightarrow{\;T\;} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \\ \dot{\frac{\partial \lambda}{\partial x}} \\ \dot{\frac{\partial \theta}{\partial x}} \\ \dot{\frac{\partial \lambda}{\partial y}} \\ \dot{\frac{\partial \theta}{\partial y}} \end{bmatrix} \xrightarrow{\begin{bmatrix} T & & \\ \begin{matrix} DX \cdot A \\ DY \cdot B \end{matrix} & T & \\ \begin{matrix} DY \cdot A \\ DY \cdot B \end{matrix} & & T \end{bmatrix}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\frac{\partial x}{\partial x}} \\ \dot{\frac{\partial y}{\partial x}} \\ \dot{\frac{\partial x}{\partial y}} \\ \dot{\frac{\partial y}{\partial y}} \end{bmatrix}$$

where

$$A = \begin{bmatrix} \frac{\partial^2 x}{\partial \lambda^2} & \frac{\partial^2 x}{\partial \lambda \partial \theta} \\ \frac{\partial^2 x}{\partial \lambda \partial \theta} & \frac{\partial^2 x}{\partial \theta^2} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\partial^2 y}{\partial \lambda^2} & \frac{\partial^2 y}{\partial \lambda \partial \theta} \\ \frac{\partial^2 y}{\partial \lambda \partial \theta} & \frac{\partial^2 y}{\partial \theta^2} \end{bmatrix}$$

and the second derivatives transform as:

$$\begin{bmatrix} \frac{\partial^2 p}{\partial x^2} \\ \frac{\partial p}{\partial x} \\ \frac{\partial^2 p}{\partial x \partial y} \\ \frac{\partial p}{\partial y} \\ \frac{\partial^2 p}{\partial y^2} \end{bmatrix} \xrightarrow{K} \begin{bmatrix} \frac{\partial^2}{\partial \lambda^2} \\ \frac{\partial p}{\partial \lambda} \\ \frac{\partial^2 p}{\partial \lambda \partial \theta} \\ \frac{\partial p}{\partial \theta} \\ \frac{\partial^2 p}{\partial \theta^2} \end{bmatrix} \qquad \begin{bmatrix} \dot{\lambda}^2 \\ \ddot{\lambda} \\ 2\,\dot{\lambda}\dot{\theta} \\ \ddot{\theta} \\ \dot{\theta}^2 \end{bmatrix} \xrightarrow{U} \begin{bmatrix} \dot{x}^2 \\ \ddot{x} \\ 2\,\dot{x}\dot{y} \\ \ddot{y} \\ \dot{y}^2 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial^2 \dot{\lambda}}{\partial \lambda^2} \\ \frac{\partial \dot{\lambda}}{\partial \lambda} \\ \frac{\partial^2 \dot{\lambda}}{\partial \lambda \partial \theta} \\ \frac{\partial \dot{\lambda}}{\partial \theta} \\ \frac{\partial^2 \dot{\lambda}}{\partial \theta^2} \\ \frac{\partial^2 \dot{\theta}}{\partial \lambda^2} \\ \frac{\partial \dot{\theta}}{\partial \lambda} \\ \frac{\partial^2 \dot{\theta}}{\partial \lambda \partial \theta} \\ \frac{\partial \dot{\theta}}{\partial \theta} \\ \frac{\partial^2 \dot{\theta}}{\partial \theta^2} \end{bmatrix} \xrightarrow{\begin{bmatrix} K^{-1} & 0 \\ 0 & K^{-1} \end{bmatrix}} \begin{bmatrix} \frac{\partial^2 \dot{\lambda}}{\partial x^2} \\ \frac{\partial \dot{\lambda}}{\partial x} \\ \frac{\partial^2 \dot{\lambda}}{\partial x \partial y} \\ \frac{\partial \dot{\lambda}}{\partial y} \\ \frac{\partial^2 \dot{\lambda}}{\partial y^2} \\ \frac{\partial^2 \dot{\theta}}{\partial x^2} \\ \frac{\partial \dot{\theta}}{\partial x} \\ \frac{\partial^2 \dot{\theta}}{\partial x \partial y} \\ \frac{\partial \dot{\theta}}{\partial y} \\ \frac{\partial^2 \dot{\theta}}{\partial y^2} \end{bmatrix}$$

$$
\begin{bmatrix} \dot{\lambda} \\ \dot{\theta} \\ \frac{\partial \dot{\lambda}}{\partial x} \\ \frac{\partial \dot{\theta}}{\partial x} \\ \frac{\partial \dot{\lambda}}{\partial y} \\ \frac{\partial \dot{\theta}}{\partial y} \\ \frac{\partial^2 \dot{\lambda}}{\partial x^2} \\ \frac{\partial^2 \dot{\theta}}{\partial x^2} \\ \frac{\partial^2 \dot{\lambda}}{\partial y^2} \\ \frac{\partial^2 \dot{\theta}}{\partial y^2} \\ \frac{\partial^2 \dot{\lambda}}{\partial x \partial y} \\ \frac{\partial^2 \dot{\theta}}{\partial x \partial y} \end{bmatrix}
\begin{bmatrix}
T & & & & & \\
\begin{matrix} DX \cdot A \\ DX \cdot B \end{matrix} & T & & & & \\
\begin{matrix} DY \cdot A \\ DY \cdot B \end{matrix} & O & T & & & \\
\begin{matrix} d_x \cdot C \\ d_x \cdot D \end{matrix} & \begin{matrix} 2DX \cdot A \\ 2DX \cdot B \end{matrix} & O & T & & \\
\begin{matrix} d_y \cdot C \\ d_y \cdot D \end{matrix} & O & \begin{matrix} 2DY \cdot A \\ 2DY \cdot B \end{matrix} & O & T & \\
\begin{matrix} d_{xy} \cdot C \\ d_{xy} \cdot D \end{matrix} & \begin{matrix} DY \cdot A \\ DY \cdot B \end{matrix} & \begin{matrix} DX \cdot A \\ DX \cdot B \end{matrix} & O & O & T
\end{bmatrix}
\longrightarrow
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \frac{\partial \dot{x}}{\partial x} \\ \frac{\partial \dot{y}}{\partial x} \\ \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial y} \\ \frac{\partial^2 \dot{x}}{\partial x^2} \\ \frac{\partial^2 \dot{y}}{\partial x^2} \\ \frac{\partial^2 \dot{x}}{\partial y^2} \\ \frac{\partial^2 \dot{y}}{\partial y^2} \\ \frac{\partial^2 \dot{x}}{\partial x \partial y} \\ \frac{\partial^2 \dot{y}}{\partial x \partial y} \end{bmatrix}
$$

$$C \equiv \begin{bmatrix} \frac{\partial^3 x}{\partial \lambda^3} & \frac{\partial^3 x}{\partial^2 \lambda \partial \theta} \\ \frac{\partial^2 x}{\partial \lambda^2} & \frac{\partial^2 \chi}{\partial \lambda \partial \theta} \\ \frac{\partial^3 x}{\partial^2 \lambda \partial \theta} & \frac{\partial^3 x}{\partial \lambda \partial \theta^2} \\ \frac{\partial^2 x}{\partial \lambda \partial \theta} & \frac{\partial^2 x}{\partial \theta^2} \\ \frac{\partial^3 x}{\partial \lambda \partial^2 \theta} & \frac{\partial^3 x}{\partial \theta^3} \end{bmatrix}$$

$$D \equiv \begin{bmatrix} \frac{\partial^3 y}{\partial \lambda^3} & \frac{\partial^3 y}{\partial^2 \lambda \partial \theta} \\ \frac{\partial^2 y}{\partial \lambda^2} & \frac{\partial^2 y}{\partial \lambda \partial \theta} \\ \frac{\partial^3 y}{\partial^2 \lambda \partial \theta} & \frac{\partial^3 y}{\partial \lambda \partial^2 \theta} \\ \frac{\partial^2 y}{\partial \lambda \partial \theta} & \frac{\partial^2 y}{\partial \theta^2} \\ \frac{\partial^3 y}{\partial \lambda \partial^2 \theta} & \frac{\partial^3 y}{\partial \theta^3} \end{bmatrix}$$

Finally, as a comprehensive test of the program, a multilevel run was made with T first maintained as a constant. We should find

1a) $\omega = 0$

1b) $\overline{\omega} = -\frac{\sigma}{P_*}\left\{\frac{\partial P_*}{\partial t} + D(1) - P_* D(P_*^{-1})\right\}$

1c) $D(1) = \frac{\partial}{\partial\sigma}\left\{\sigma\ [\overline{D}(1) - P_* D(P_*^{-1})]\right\}$

$$\text{with } D(\ ) = \frac{1}{a\cos\theta}\left[\frac{\partial}{\partial\lambda} P_* u(\ ) + \frac{\partial}{\partial\theta} P_* v\cos\theta(\ )\right]$$

The test agreed with these results. Next, the model was run with ω maintained as zero, and it was found:

2a) Same as 1b).

2b) Same as 1c).

2c) $\frac{\partial T}{\partial t} \neq 0$ as expected.

Finally, when the model was run with $\overline{\omega}$ maintained as zero, it was found:

3a) $\omega = -\sigma P_* D(P_*^{-1})$

3b) $\frac{\partial D(1)}{\partial\sigma} = 0$

3c) $\frac{\partial T}{\partial t}$ is different than that found in 2c).

V. Filter

The filter applied east-west is

$$f_i^{(p+1)} = \left[1 + (-1)^p\ (\delta/2)^{2p+2}\right] f_i \equiv F^{EW} f_i$$

where f_i is the current variable field, and

$$\delta^2 f_i = f_{i-1} - 2f_i + f_{i+1}$$

The filter applied north-south is:

$$g_j^{(p+1)} = \left[1 + (-1)^p \left(\delta/2\right)^{2p+2} \right] g_j \equiv F^{NS} \left(F^{EW} f_i\right)$$

where g_j is the field coming out of the east west filter, and:

$$\cos\theta_j \, \delta^2 g_j = (f_{j-1} - f_j) \cos\theta_{j-\frac{1}{2}} - (f_j - f_{j+1}) \cos\theta_{j+\frac{1}{2}}$$

The above filter was applied with $p = 3$, both east-west and north-south on a 16 x 32 grid. F^{NS} does not eliminate the effect of the 2-grid interval wave. In fact, if $\vec{v}$ is the two grid interval wave:

$$F^{NS} \vec{v} = (1 - \cos^4\alpha) \vec{v}$$

where $\alpha = \Delta\theta/2$.

Thus $F^{NS} \vec{v} \approx .019\vec{v}$. The Legendre polynomials $[1]$ and $[\sin\theta]$ are also eigenvectors of F^{NS}; in fact:

$$F^{NS}[1] = [1]$$

$$F^{NS}[\sin\theta] = \left\{ 1 - \frac{1}{256} \lambda^4 \right\} [\sin\theta]$$

where $\lambda = -2(\cos\alpha - \cos3\alpha)$

Method

After the completion of a regular GEX time step, NTS is tested to determine if the current time step is to be filtered. If so, all the IOT data is brought band by band back into BLK (each record holding 384 words). After this data is filtered, it replaces the original data on IOT, then, if IN1 data (NTS-1) was not filtered previously, it is filtered now.

For a detailed description of the filter see the Appendix D.

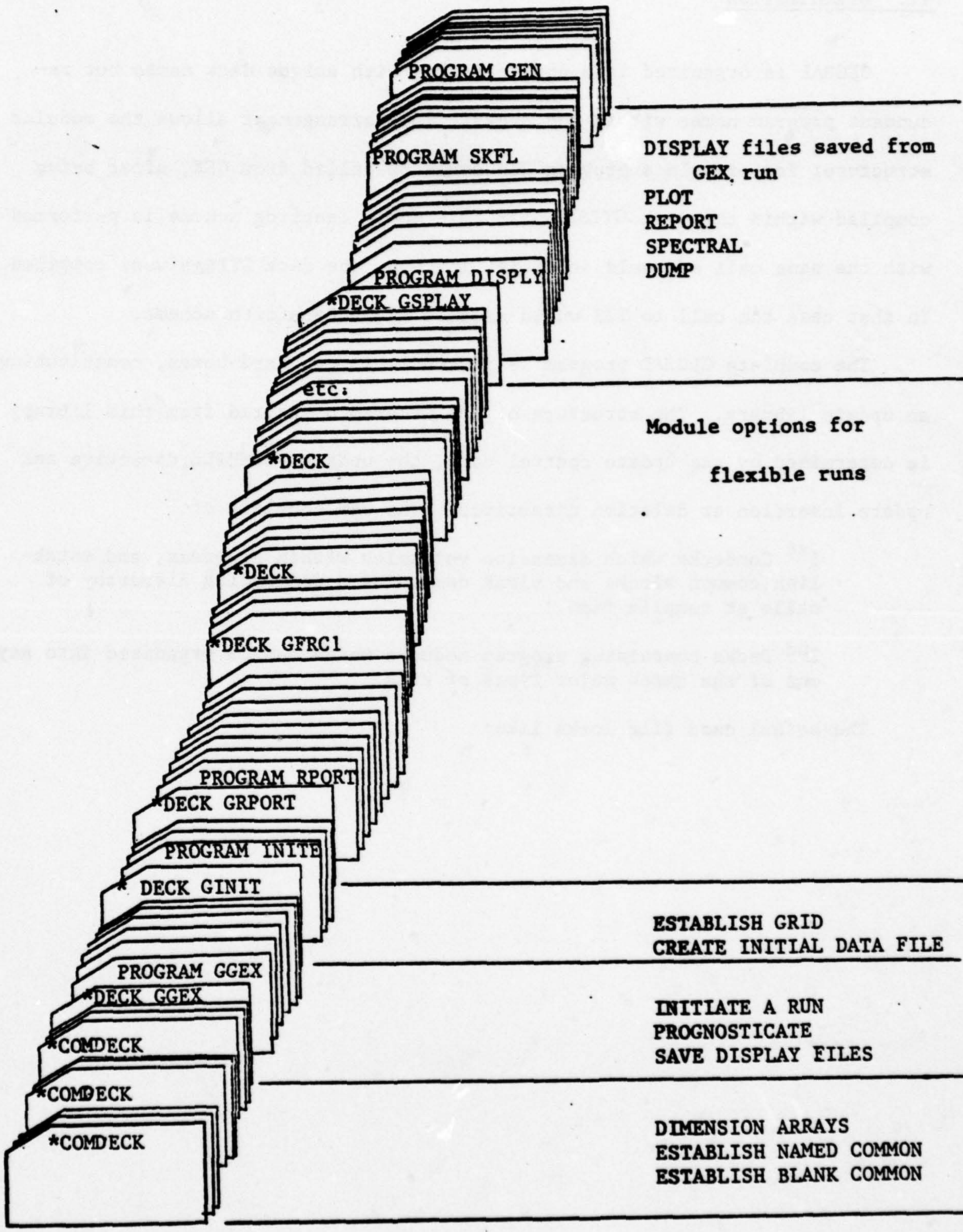
VI. Organization

GLOBAL is organized into update decks, with unique deck names but redundant program names within the decks. This arrangement allows the modular structure; for example a program TIS could be called from GEX, after being compiled within the deck GTISLF. In this way a leapfrog scheme is performed with the same call as would occur if, instead, the deck GTISAB were compiled. In that case the call to TIS would execute Adams Bashforth scheme.

The complete GLOBAL program is presently in two card boxes, constituting an update library. The structure of the program organized from this library is determined by the update control card, the update *COMPILE directive and update insertion or deletion directives. The box consists of:

> 1st Comdecks which dimension variables within programs, and establish common blocks and blank common by a descending hierarchy of calls at compile time.
>
> 2nd Decks containing program modules which may be organized into any one of the three major types of runs.

The actual card file looks like:


PROGRAM GEN
PROGRAM SKFL
PROGRAM DISPLY
*DECK GSPLAY
etc.
*DECK
*DECK
*DECK GFRC1
PROGRAM RPORT
*DECK GRPORT
PROGRAM INITE
* DECK GINIT
PROGRAM GGEX
*DECK GGEX
*COMDECK
*COMDECK
*COMDECK
DISPLAY files saved from GEX run
PLOT
REPORT
SPECTRAL
DUMP
Module options for flexible runs
ESTABLISH GRID
CREATE INITIAL DATA FILE
INITIATE A RUN
PROGNOSTICATE
SAVE DISPLAY FILES
DIMENSION ARRAYS
ESTABLISH NAMED COMMON
ESTABLISH BLANK COMMON

VII. Macro Flowchart

INITE → GEX → DISPLY

INITE: INITE is a self contained program to generate the data file and the grid conditions for a run.

GEX: Is the solution program, which accepts initial data and a defining grid from INITE, and steps the fields forward in time. GEX also saves user specified time steps for later display. Optionally, GEX may accept two contiguous time steps from an interrupted run, and continue that run. GEX stands for Global Experiment.

DISPLY: DISPLY is a self contained program which will analyze and report the data generated by the solution program GEX.

In the following we shall describe all three modules. However, since many of the variables appear in two or all three, we shall describe a variable only once, the first time it appears, to avoid redundancy. Often this will result in describing variables that appear, but are not used in a subroutine, and are only used later.

APPENDICES:

GLOBAL

DOCUMENTATION

APPENDIX A: INITE

Characteristics:

Core: 64000_B

Time: 20 seconds should be enough

Options: May choose display of initial data field and grid on community output file

Permanent Files: Initial data file
Grid file

Organization of Core During a Run

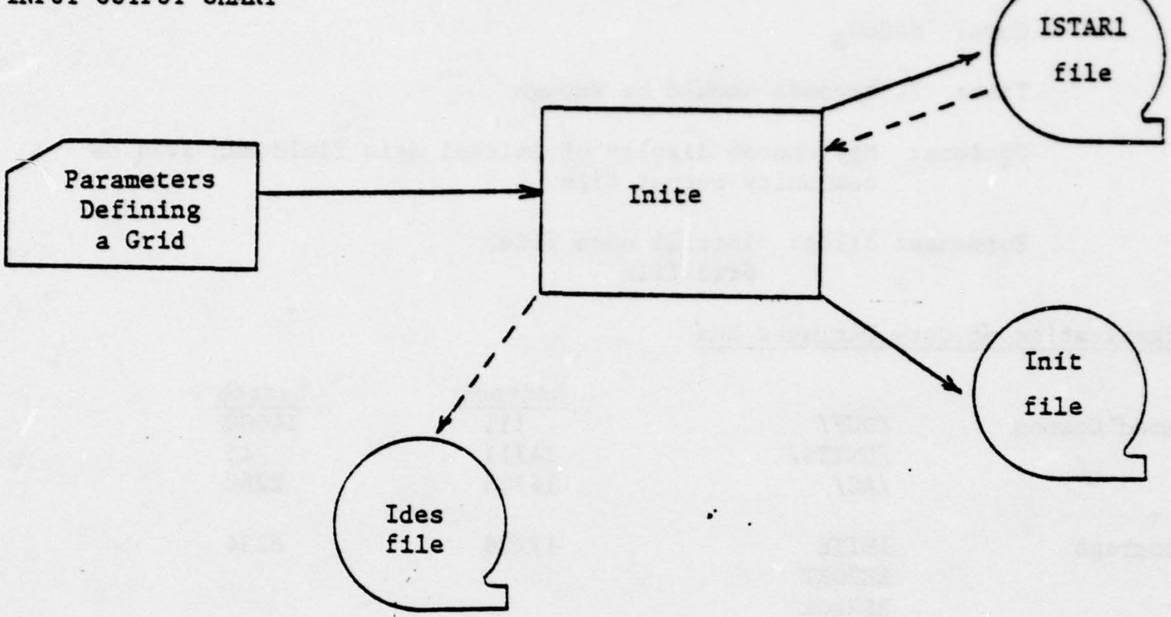
		Address	Length
Named Common	/BUF/	111	14600
	/UNITS/	14711	15
	/AC/	14726	2260
Programs	INITE	17206	6234
	REPORT		
	BERROR		
	OBAL		
	COMP		
	SLAY		
	RINT		
	PRTR		
	PRTI		
	PRTR2	30531	174
	CBUFN		
SYSTEM			
Blank Common	/ /	47552	12000

Common Blocks

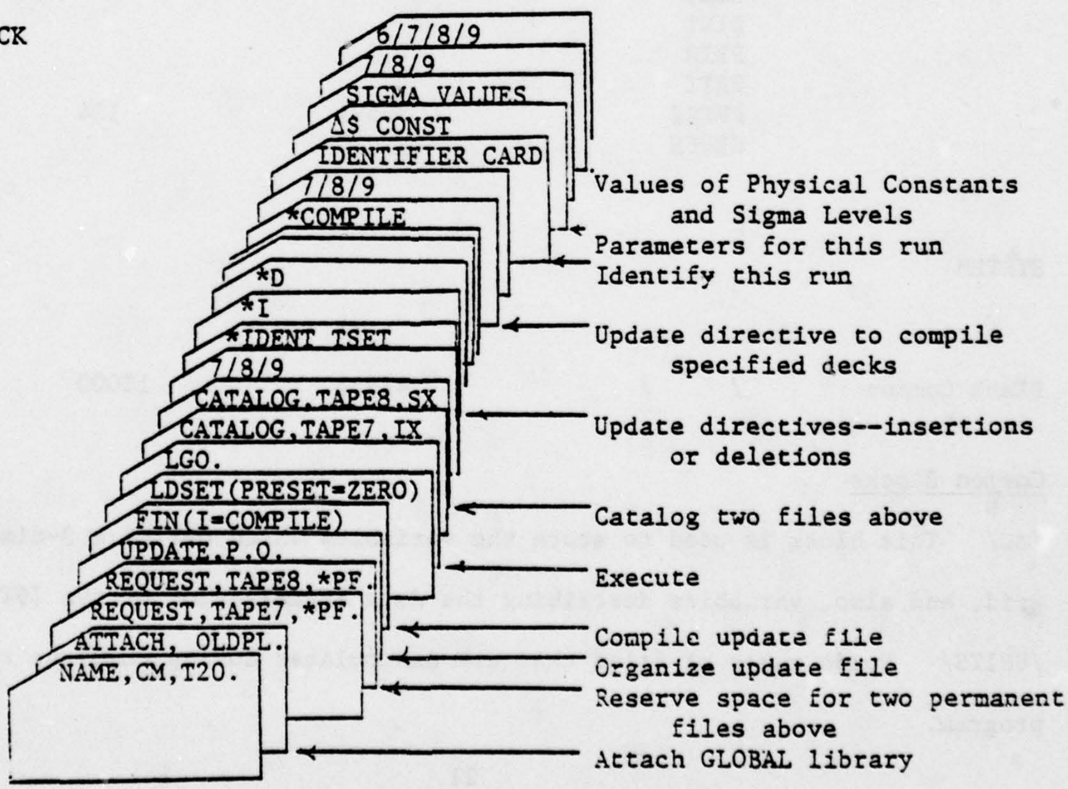
/AC/ This block is used to store the variables which define a 3-dimensional grid, and also, variables describing the data organization of the ISTAR 1 file.

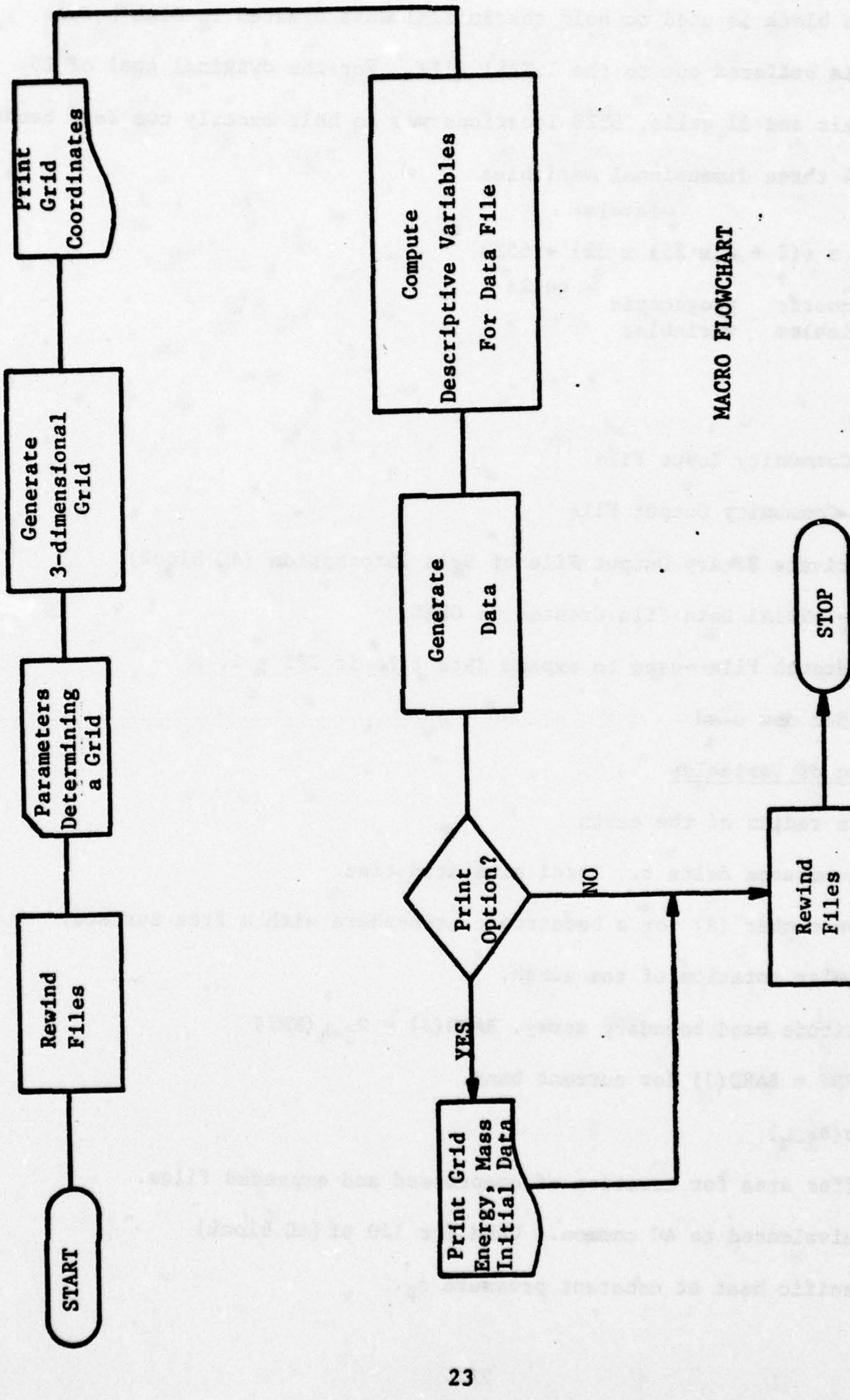
/UNITS/ Holds names of files that are manipulated during complete run of program.

INPUT OUTPUT CHART



SAMPLE DECK





MACRO FLOWCHART

/BUF/ This block is used to hold the initial data created in OBAL before it is buffered out to the ISTAR1 file. For the original goal of 25 levels and 32 cells, 6528 locations was to hold exactly two data bands of 4 three dimensional variables

$$2 \times \{(\underbrace{2}_{\text{diagnostic variables}} + \underbrace{4}_{\text{prognostic variables}} \times \overbrace{25}^{\text{levels}}) \times \underbrace{32}_{\text{cells}}\} = 6528$$

<u>Files</u>

INPUT(5)--Community Input File

OUTPUT(6)--Community Output File

INIT(7)--Private Binary Output File of Grid Information (AC block)

ISTAR1(8)--Initial Data File Created in OBAL

IDES(9)--Scratch File--used to expand data file if IPT = 1.

(10)--Not now used

<u>Description of Variables</u>

A	Mean radius of the earth
ADT	Accumulated delta t. Total simulated time.
ARE	Wave number (R) for a baratropic atmosphere with a free surface.
AVRE	Angular rotation of the earth.
BAND	Latitude band boundary array, BAND(J) = $\theta_{j-\frac{1}{2}}$(KMI)
BANDJ	BANDJ = BAND(J) for current band.
BCTHJ	$\cos(\theta_{j-\frac{1}{2}})$
BLK	Buffer area for creation of compressed and expanded files.
CBLK	Equivalenced to AC common. Used for I/O of (AC block)
CP	Specific heat at constant pressure c_p.

CTHJ	Array holding $\cos(\theta_j)$
DEV	Array holding deviations of cell (1,j) from the zero meridian in fractions of $\Delta\lambda_j$. If γ_j is the angle of deviation, then $DEV(J) = \alpha_j = \gamma_j/\Delta\lambda_j$ (KMI).
DL	Array of zonal angle increments $DL(J) = \Delta\lambda_j$.
DLL	Meridional angle increment $DLL = \theta_{j-\frac{1}{2}} - \theta_{j+\frac{1}{2}}$.
DT	Simulated time increment Δt. Value is read in as parameter on entering GEX.
DTD	$2\Delta t$
DTH	$\Delta t/2$
DTHJ	$\frac{1}{2}(\theta_{j-\frac{1}{2}} - \theta_{j+\frac{1}{2}}) = \Delta\theta_j/2 \equiv \alpha$
DTI	Duration of the time integration in simulated time.
DWNS	Array of $\Delta W = W_N - W_S$, where $W_N = \sum_N W_l$ and $W_S = \sum_S W_l$; then $DWNS(J) = \Delta W_j$. Used in calculation of G_θ operator (KH).
DX	Array of cell lengths at band boundaries. $DX(J) = a\cos(\theta_{j-\frac{1}{2}})\,\Delta\lambda_j$. This is a temporary array and is <u>not</u> stored on the INIT tape of grid information.
DXG	Array of cell lengths at grid points. $DXG(J) = a\cos(\theta_j)\,\Delta\lambda_j$. This is a temporary array and is <u>not</u> stored on the INIT tape of grid information.
DY	Array of widths of cells on band j. $DY(J) = a\Delta\theta_j$. This is a temporary array and is <u>not</u> stored on the INIT tape of grid information.
FIT	Fixed interval of time. Simulated time increment Δt in GEX run.
FNB	Array of floated values of b_j.
GΦ	Gravitational constant g.
HFT	Array of values of $\hat{f} = \frac{1}{\Delta V}\iiint_{\Delta V} f\, dV = 2\Omega\sin\theta_{ij}\cos(\Delta\theta_{ij}/2)$ (KH).
HMT	Array of metric term coefficients. $\hat{m} = \frac{1}{\Delta V}\iiint_{\Delta V} \frac{\tan\theta}{a}\, dV = \tan\theta_{ij}/a$ (KH).

IDC 60 character identifier for INITE run.

IDES File name for display file catalogued as a permanent file in GEX run. Used only as a scratch file in an INITE run.

IDESF Frequency of display file generation in a GEX run.

JMAX Zonal index of maximum advecting velocity.

INIT Private binary output file of grid information saved in AC block. Catalogued as a permanent file in an INITE run.

INPUT Community input file.

IN1 Expanded active data file for time step N.

IN2 Expanded active data file for time step N - 1

IOSZ Size of I/O buffer BLK. Presently holds 2 full bands of data or 6528 words.

IOT Expanded active data file for time step N + 1.

IPRT Community output file.

IPT Print option for INITE run.

  IPT = 0 No point
  IPT = 1 Print all grid information and data

ISC Scratch file. Not used.

ISER Equivalenced to BLK(1). Used to pass serial number in integer format off label.

ISS Code for abnormal stop if $P_* < 0$.

  ISS = 0 No stop
  ISS = 1 Stop

ISTAR1 Initial data file for INITE run. Initial data or restart data file for subsequent GEX run.

ISTAR2 Restart data file for GEX run.

ITS Equivalenced to BLK(2). Used to pass time step number in integer format off label.

I1 INIT file name. Avoids technicality that can't pass file name in common to unit function.

J Latitude band index.

JB	Number of data bands in grid. Set as one of the parameters read in INITE run.
JB1	JB + 1. Actual index of last data band, since first data band at North Pole is a dummy.
JB1Z	JB1. Avoids use of JB in common as control on Do loop.
JB2	JB + 2. Index of first dummy band at South Pole.
JB3	JB + 3. Index of second dummy band at South Pole.
JB4	JB + 4. Index of third dummy band at South Pole.
JB5	JB + 5. Index of last dummy band at South Pole.
JLI	Array of indices of the last bands fit into the blocks of the compressed data file.
JMAX	Latitude index of the maximum advecting velocity.
JMIN	Latitude index of minimum space increment.
JX	Maximum number of data bands that can be accomodated.
K	Vertical level index.
KMAX	Vertical index of maximum advecting velocity.
KMIN	Level index of minimum space increment.
KV	Number of vertical levels.
KVM	KV - 1.
KVZ	KV + 1. For KV levels, there are KV1 boundaries.
KX	Maximum number of levels that can be accomodated. This is limited by the second dimension of the P arrays. Presently, P(4, 3, 32), so KX = 3 - 2 = 1.
K1	Storage location in P arrays for diagnostic variables and grid information.
K2	Storage location in P arrays for diagnostic variables and grid information.
L	Index for arrays in INITE run.
LD	Smallest time step number on a display file. Used to seek file in display run.

LIM Largest time step number on a display file. Used to seek file in display run.

LLR Number of cells on each band (fixed), or, if LLR = 0 code to indicate that a variable number of cells on each band (b_j) must be read in.

LLT Accumulated number of words of data, up to maximum of I/O size. Used when partitioning data into blocks for I/O.

LSZ Arrary of number of data words in an I/O block. $LSZ(L) = \sum_{J1}^{J2} LTI(J)$, where J1 = JLI(L - 1) and J2 = JLI(L).

LTI Array of number of data words in a band. LTI(J) = bj x {KV x MP + 2}: that is, KV levels of MP 3-dimensional variables plus 2 two dimensional variables, gives number of words required for each column; and b_j such columns around the band.

MP Number of 3-dimensional prognostic variables. Set as a read in parameter in INITE.

MTS Maximum time step desired in a GEX run.

NB Array of number of cells on each band (b_j).

$\hat{N}$BJX Maximum value in array NB above.

NBLK Number of records the compressed data has been partitioned into.

NDES Running count of the number of display files generated in a GEX run.

NS Number of data words required in a vertical column (KV x MP + 2).

NSER First word on data file label. A number used to uniquely identify INITE creation run data.

NTS Time step currently being processed during a GEX run.

R Gas constant. Don't confuse with wave number (ARE).

RTJ Real time variable.

RTL Real time limit. Used to initiate stop processing in a GEX run, if time limit is exceeded.

SIG Array of sigma values SIG(k) = $\sigma_{k-\frac{1}{2}}$ (KH)
SIG(1) = $\sigma_{\frac{1}{2}}$ top of atmosphere
SIG(KV1) = $\sigma_{kv+\frac{1}{2}}$ bottom of atmosphere

SIG1 Array of $\Delta\sigma$. SIG1(k) = $\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}$. See KH finite difference forms.

SIG2 Array of $\frac{1}{2\Delta\sigma}$. SIG2(k) = $\frac{1}{2(\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}})}$. See KH finite difference forms.

SIG3 Array SIG3(k) = $\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})$ (KH).

SIG4 Array SIG4(k) = $\frac{R\,\Delta\sigma}{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})}$ (KH).

SIG5 Array SIG5(k) = $\frac{R}{c_p\ \frac{1}{2}\ (\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})}$ (KH).

SMIN Minimum space increment - $(\Delta V_{ijk})_{min}$.

THJ Temporary variable θ_j.

T1 $\frac{1}{2}b_j/(a \cos\Theta j \sin {}^{\Delta\Theta}j/2)$. Used in the calculation of weights. (KMI).

UN Returned value of unit function.

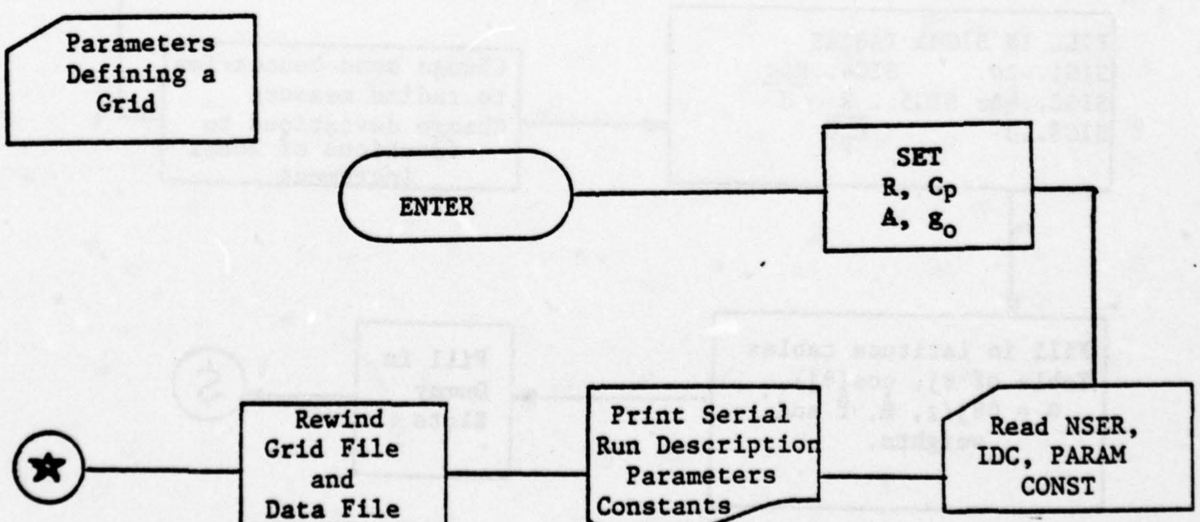
VMAX Maximum value of advecting velocity.

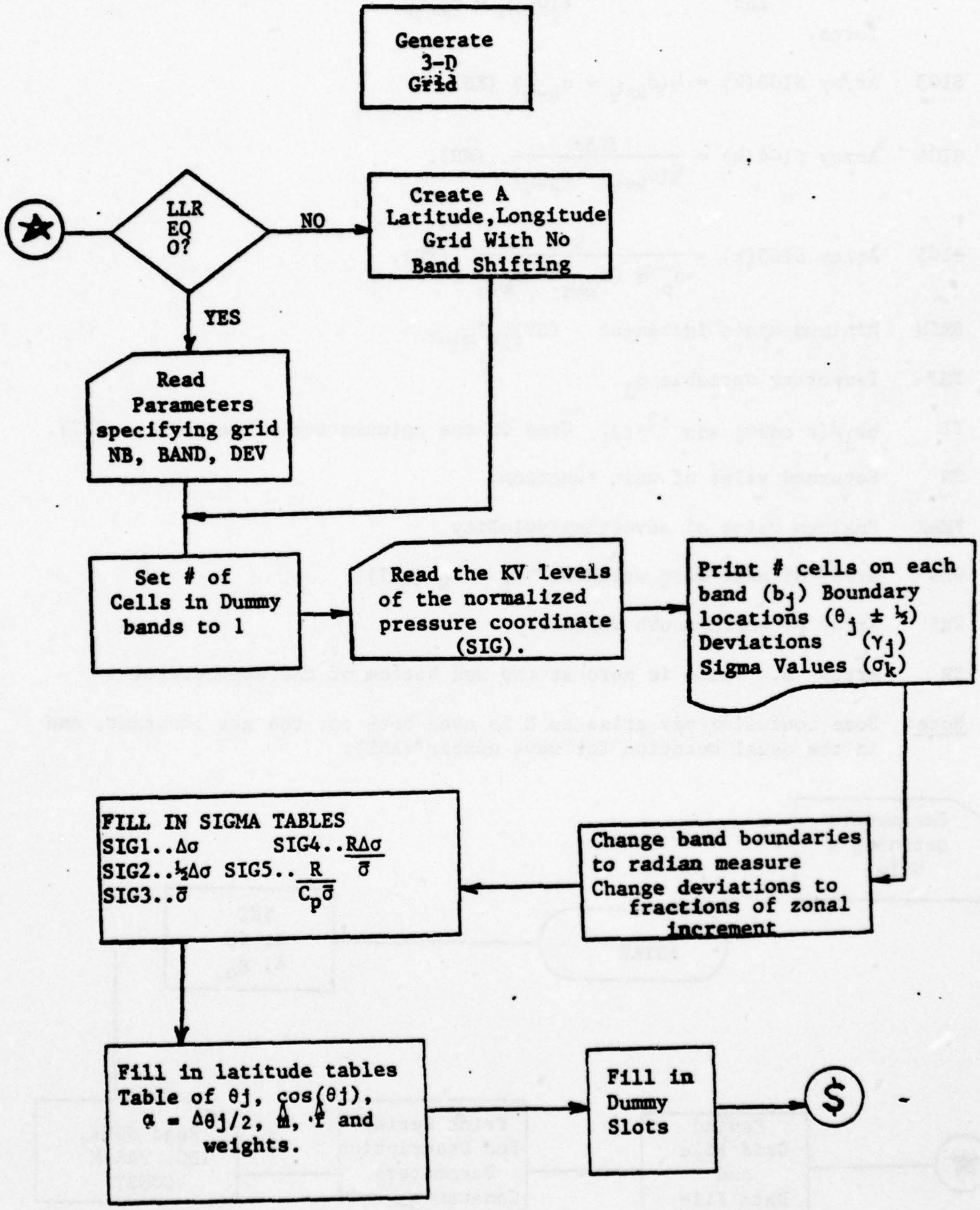
WEW Array of east-west weights. $W_E = W_W$ (KMI).

WNS Array of north-south weights.

ZB Array $\overline{\omega}$. Value is zero at top and bottom of the atmosphere.

Note: Some confusion may arise as R is used both for the gas constant, and in the usual notation for wave number (ARE).

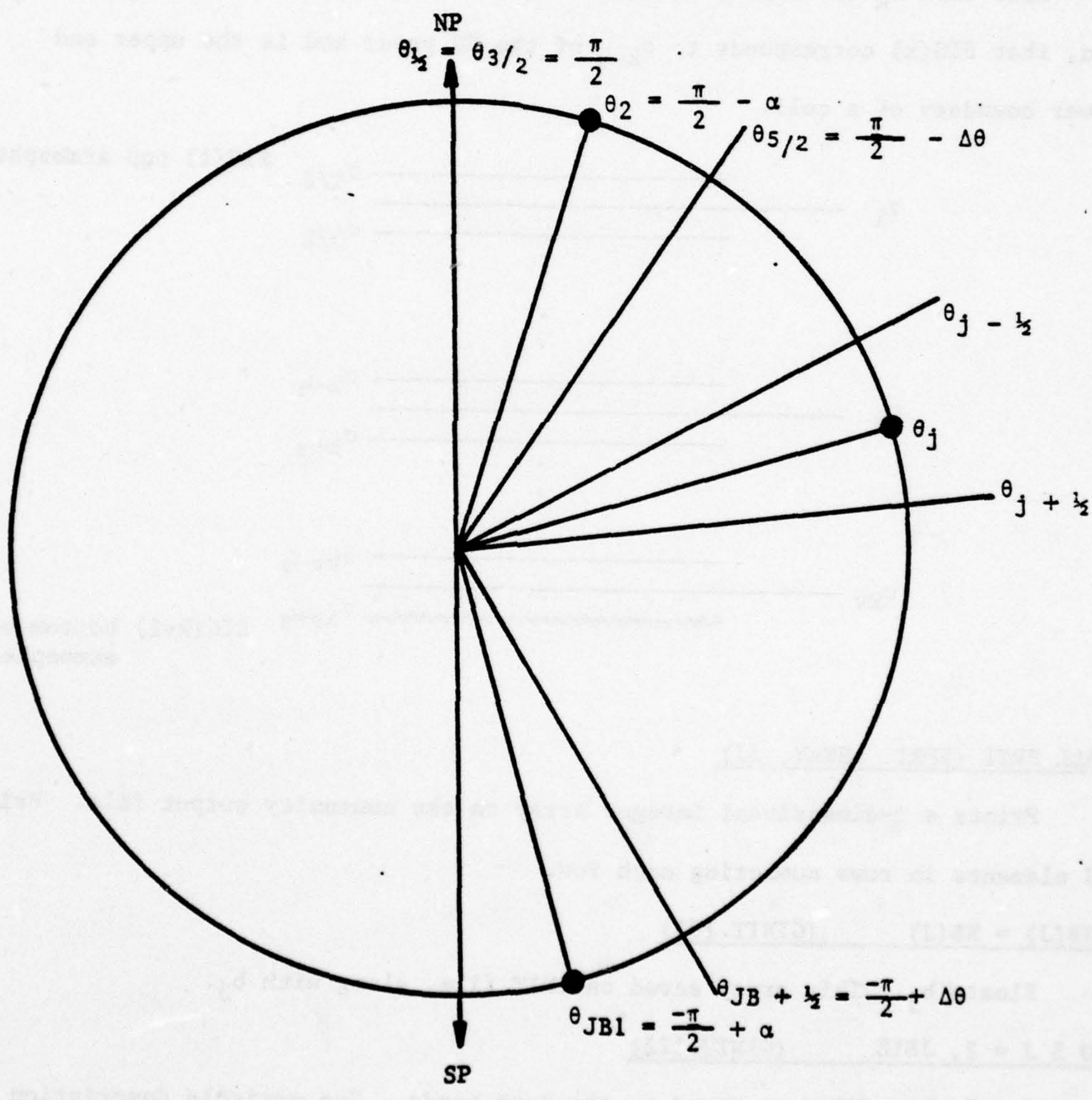



Generate
3-D
Grid
LLR
EQ
0?
NO
Create A
Latitude, Longitude
Grid With No
Band Shifting
YES
Read
Parameters
specifying grid
NB, BAND, DEV
Set # of
Cells in Dummy
bands to 1
Read the KV levels
of the normalized
pressure coordinate
(SIG).
Print # cells on each
band (b_j) Boundary
locations ($\theta_j \pm \frac{1}{2}$)
Deviations (γ_j)
Sigma Values (σ_k)
FILL IN SIGMA TABLES
SIG1..$\Delta\sigma$
SIG2..$\frac{1}{2}\Delta\sigma$
SIG3..$\bar{\sigma}$
SIG4..$\frac{R\Delta\sigma}{\bar{\sigma}}$
SIG5..$\frac{R}{C_p\bar{\sigma}}$
Change band boundaries
to radian measure
Change deviations to
fractions of zonal
increment
Fill in latitude tables
Table of θj, $\cos(\theta j)$,
$\alpha = \Delta\theta j/2$, $\hat{m}$, $\hat{f}$ and
weights.
Fill in
Dummy
Slots
$

Helpful Comments

DLL = 180./FLOAT(JB) $ BAND(1) = 90. $ BAND(2) = 90. (Ginit.61)

There is a single band at the north pole corresponding to the index J = 1. Thus the north pole is found at $\theta_{1/2} = 90$, where BAND(1) = $\theta_{1/2}$ and BAND(2) = $\theta_{3/2}$.



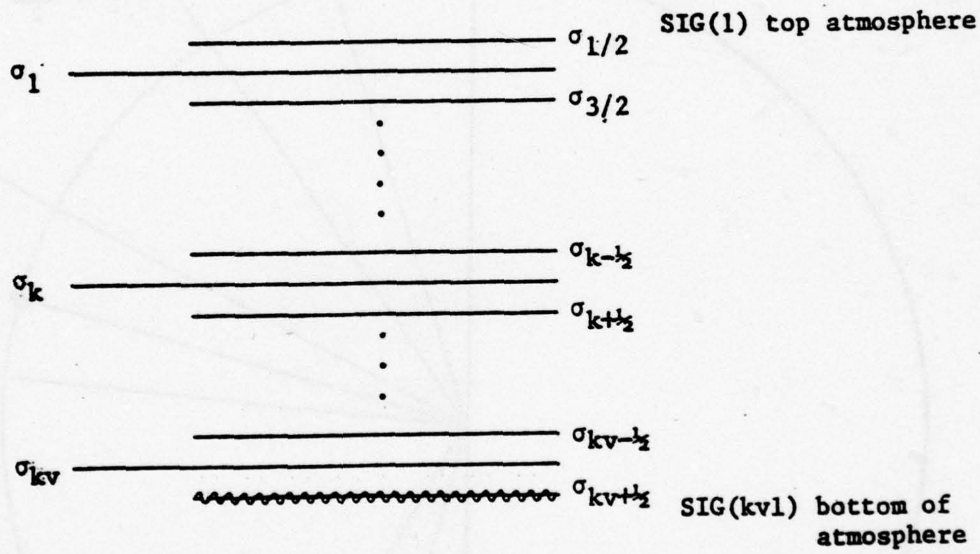
$\theta_{JB+3/2} = \theta_{JB+5/2} = \theta_{JB+7/2} = \theta_{JB+9/2} = -\pi/2$

JB1 is the index of the last data band.

JB2, JB3, JB4, JB5 correspond to dummy bands at the south pole, each having one dummy cell. (NB(1) = NB(2) = NB(3) = NB(4) = NB(5) = 1) Actually NB should be dimensioned 69 to accomodate 64 data bands.

Read (INPUT, 102) (SIG(k), k = 1, kvl) (GINIT.89)

Note that σ_k is indexed from the top to the bottom of the atmosphere, and, that SIG(k) corresponds to $\sigma_{k-\frac{1}{2}}$ of the KH paper and is the upper and lower boundary of a cell.



CALL PRTI (IPRT, ARRAY, JJ)

Prints a 1-dimensional integer array on the community output file. Prints JJ elements in rows numbering each row.

FNB(J) = NB(J) (GINIT.103)

Floats b_j. This array saved on INIT file, along with b_j.

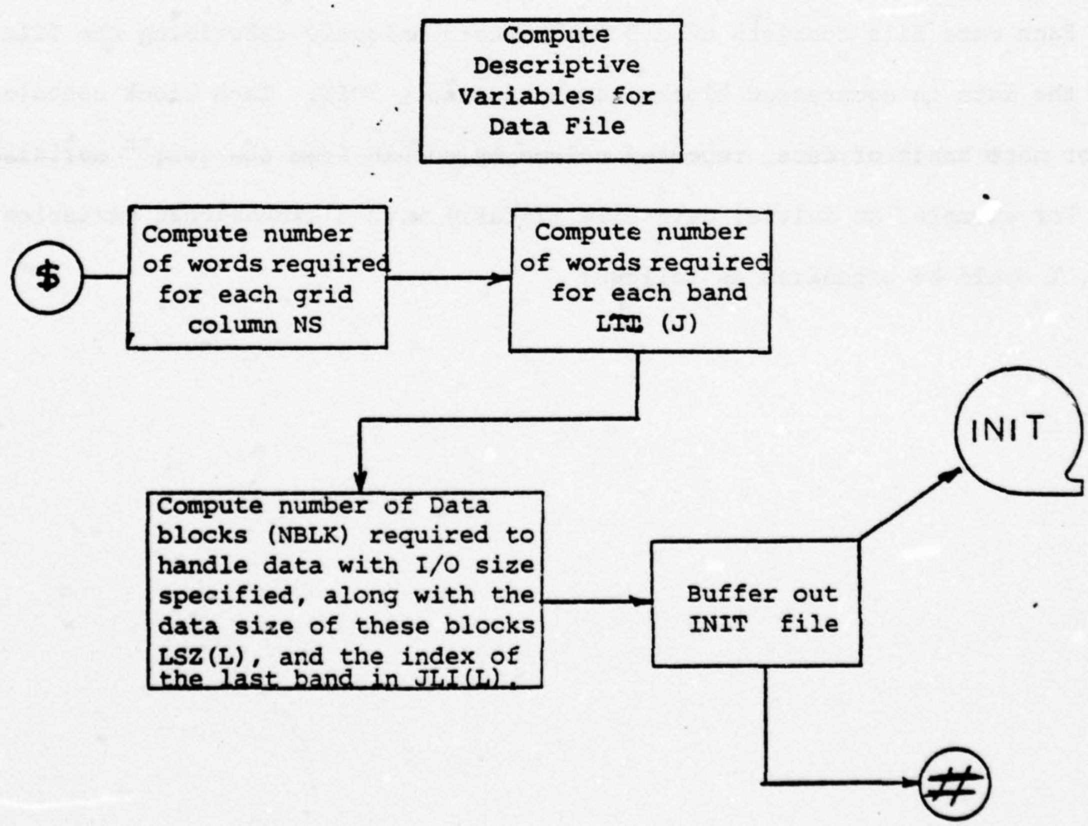
DO 5 J = 2, JBIZ (GINIT.122)

J = 2 thru JB1 correspond to the data bands. See variable description

for explanation of the arrays in this loop.

C Fill in Dummy Slots (GINIT.141)

Notice for the dummy band (J = 1), that all grid parameters, including the north boundary weights (WNS(1,1)), and the south boundary weights (WNS(1,2)) are set to zero. For the first data band (J = 2) the north boundary weight (WNS(2,1)), corresponding to the north pole, is set to zero, while the south boundary weight (WNS(2,2)), remains as it was calculated in GINIT.133. Similarly, the last data band (JB1), has its northern boundary weight set in GINIT.132, and the south boundary weight, corresponding to the south pole, set to zero (GINIT.149). For all the south polar dummy bands (JB2 - JB5), all grid parameters are set to zero.



Helpful Comments

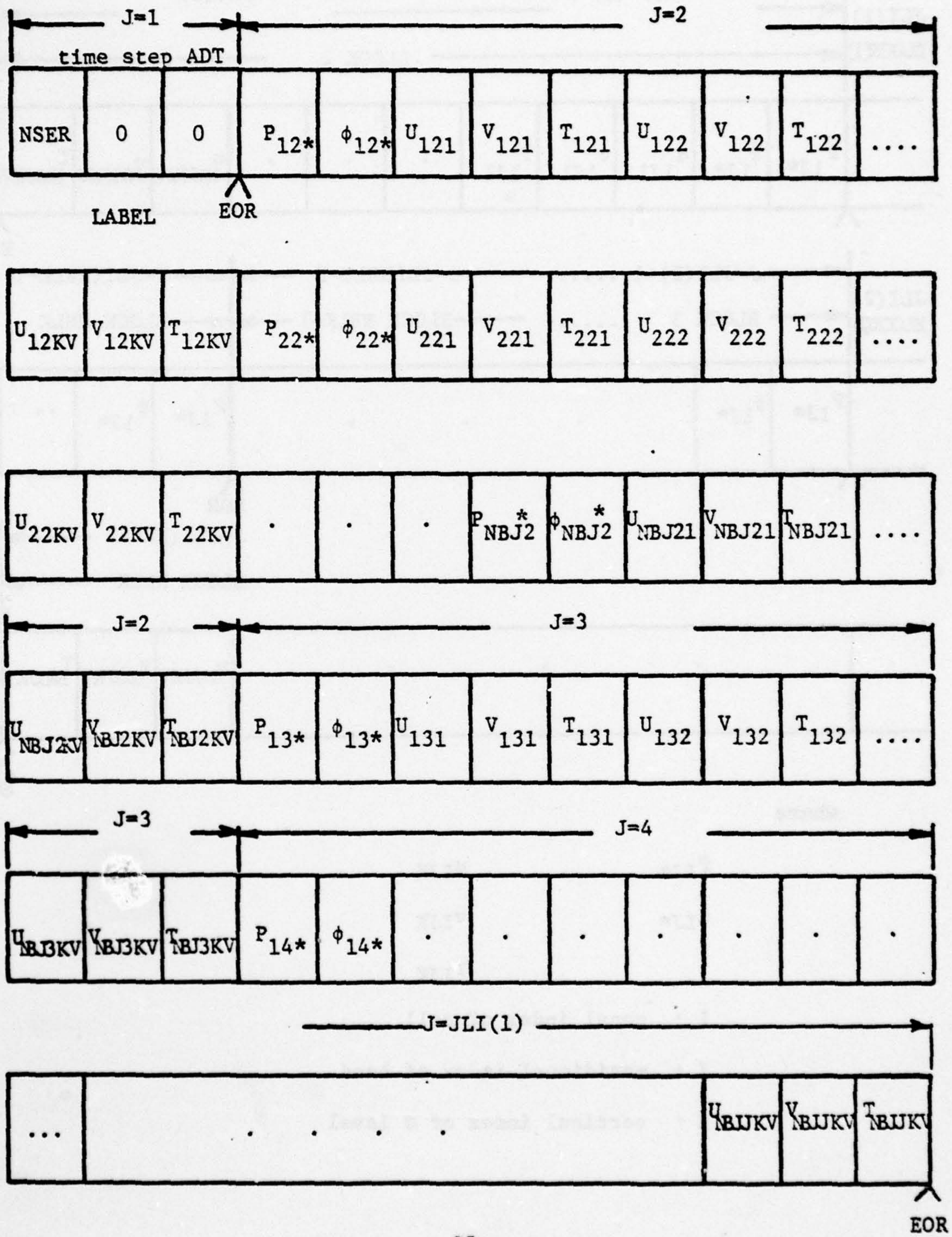
C Data Logic Section (GINIT.158)

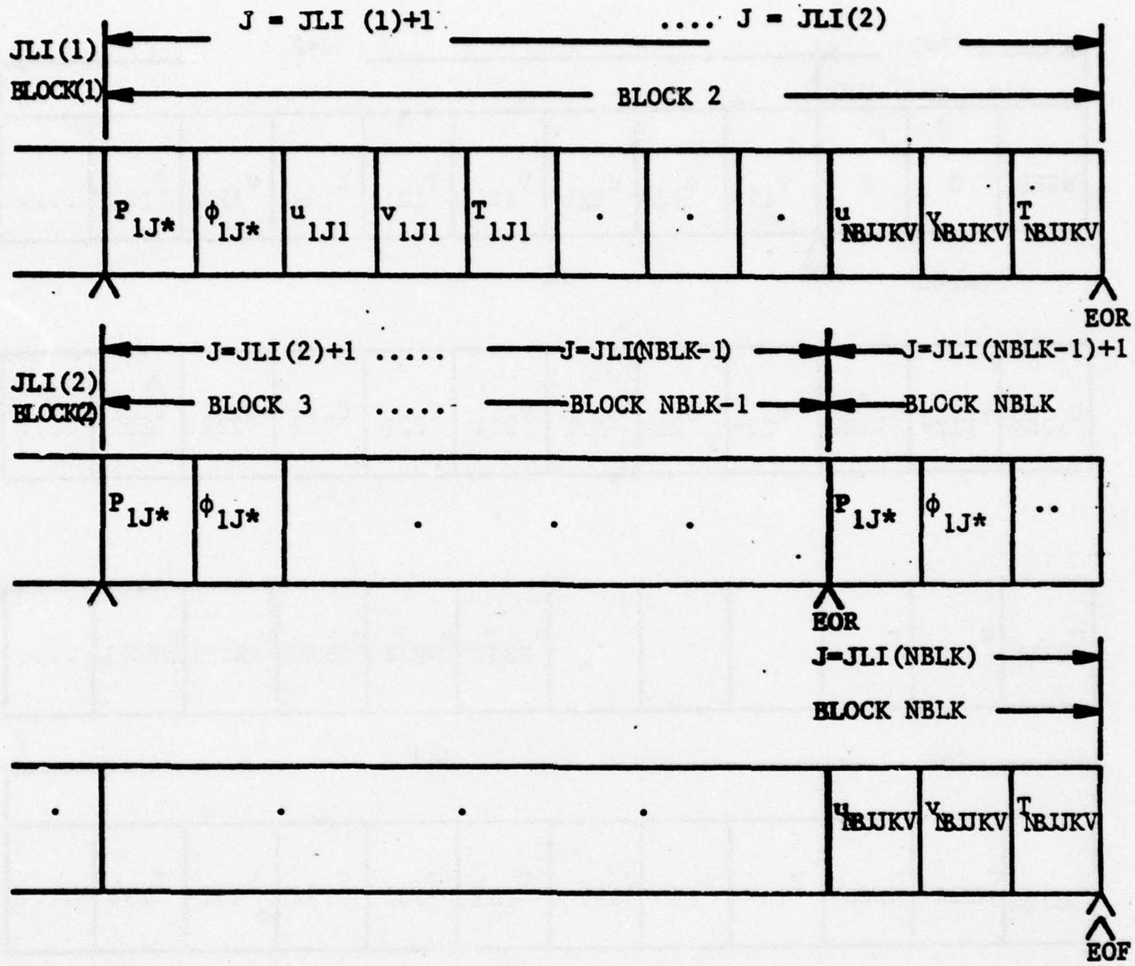
Here are created the descriptive arrays LTI, JLI, LSZ, and the parameter NBLK. These completely describe the organization of the data file ISTAR1, and all subsequent data files from GEX runs. These data files are organized in compressed format--that is only the 3-D prognostic variables, and the variables P_* and ϕ_* are carried; other variables are not carried since they can be recreated from the INIT file at the beginning of each GEX run (that is X_N', X_S', η, ξ).

Organization of the Data File

Each data file consists of a 3 work record uniquely describing the file; then the data in compressed blocks, each of size $\leq$ IOSZ. Each block contains one or more bands of data, reported column by column from the zeroth meridian.

For example, an initial data file (ISTAR1) with 3 dimensional variables u, v, T would be organized as follows:


J=1
J=2
time step ADT
NSER
0
0
P12*
ϕ12*
U121
V121
T121
U122
V122
T122
....
LABEL
EOR
U12KV
V12KV
T12KV
P22*
ϕ22*
U221
V221
T221
U222
V222
T222
U22KV
V22KV
T22KV
PNBJ2*
ϕNBJ2*
UNBJ21
VNBJ21
TNBJ21
J=2
J=3
UNBJ2KV
VNBJ2KV
TNBJ2KV
P13*
ϕ13*
U131
V131
T131
U132
V132
T132
J=3
J=4
UNBJ3KV
VNBJ3KV
TNBJ3KV
P14*
ϕ14*
J=JLI(1)
...
UNBJJKV
VNBJJKV
TNBJJKV
EOR



where

P_{IJ*} $\qquad$ u_{IJK}

ϕ_{IJ*} $\qquad$ v_{IJK}

T_{IJK}

I · zonal index of cell

J · meridional index of band

K · vertical index of σ level

NS = KV * MP + 2   (GINIT.159)

Each grid column will require NS data words on the compressed file; KV words for each of the MP 3-D variables, plus 2 more for the 2-D variables P_* and ϕ_*.

6000 LTI(J) = NB(J) * NS

Grid band J requires LTI(J) data words on the compressed file:

JLI(L) = J - 1 $ LSZ(L) = LLT - LTI(J) $ L = L + 1 $ LLT = LTI(J)

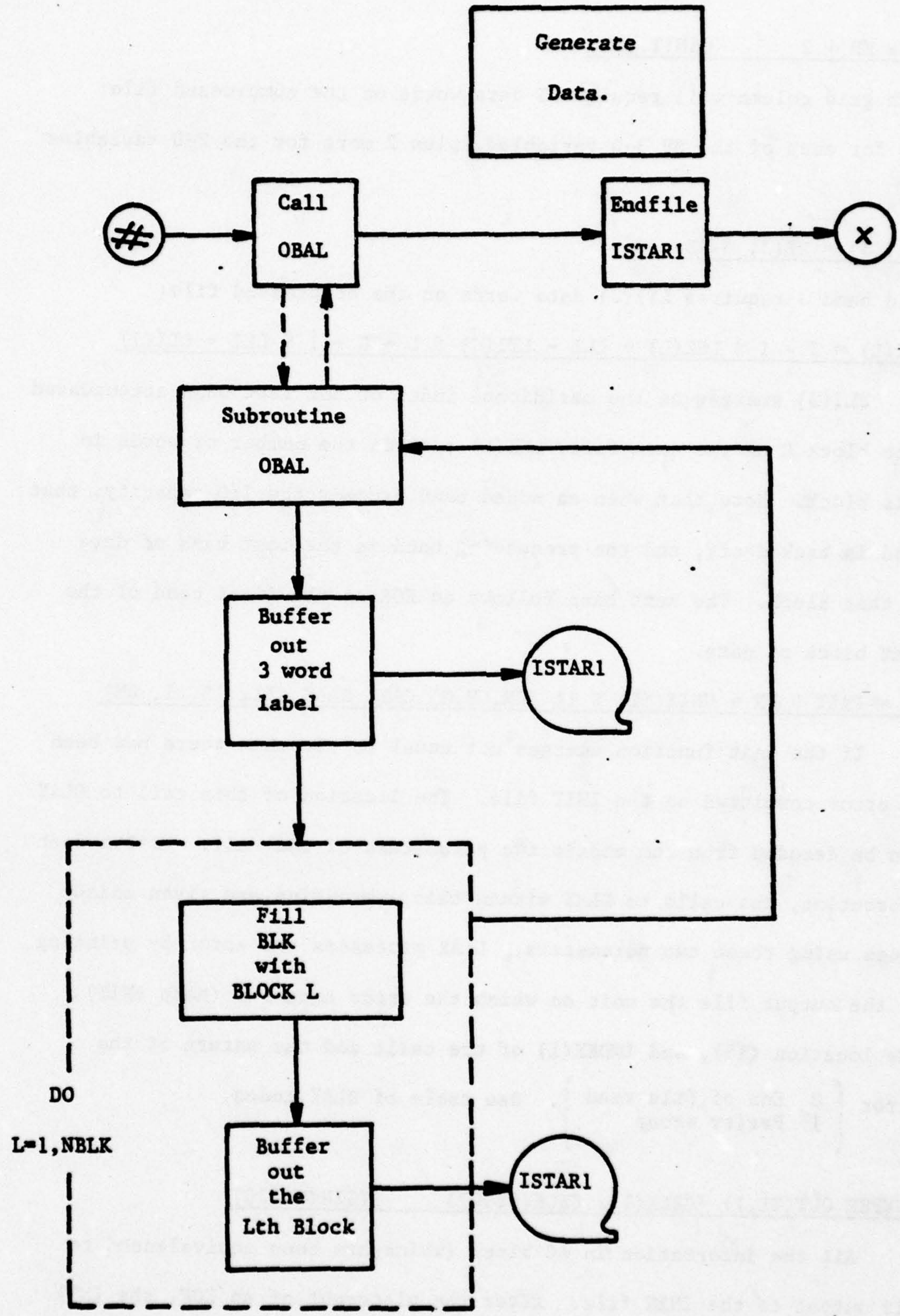
JLI(J) emerges as the meridional index of the last band accomodated into block L on the data file; LSZ(L) records the number of words in this block. Note that when an added band exceeds the I/O capacity, that band is backed off, and the preceeding band is the last band of data in that block. The next band follows an EOR as the first band of the next block of data.

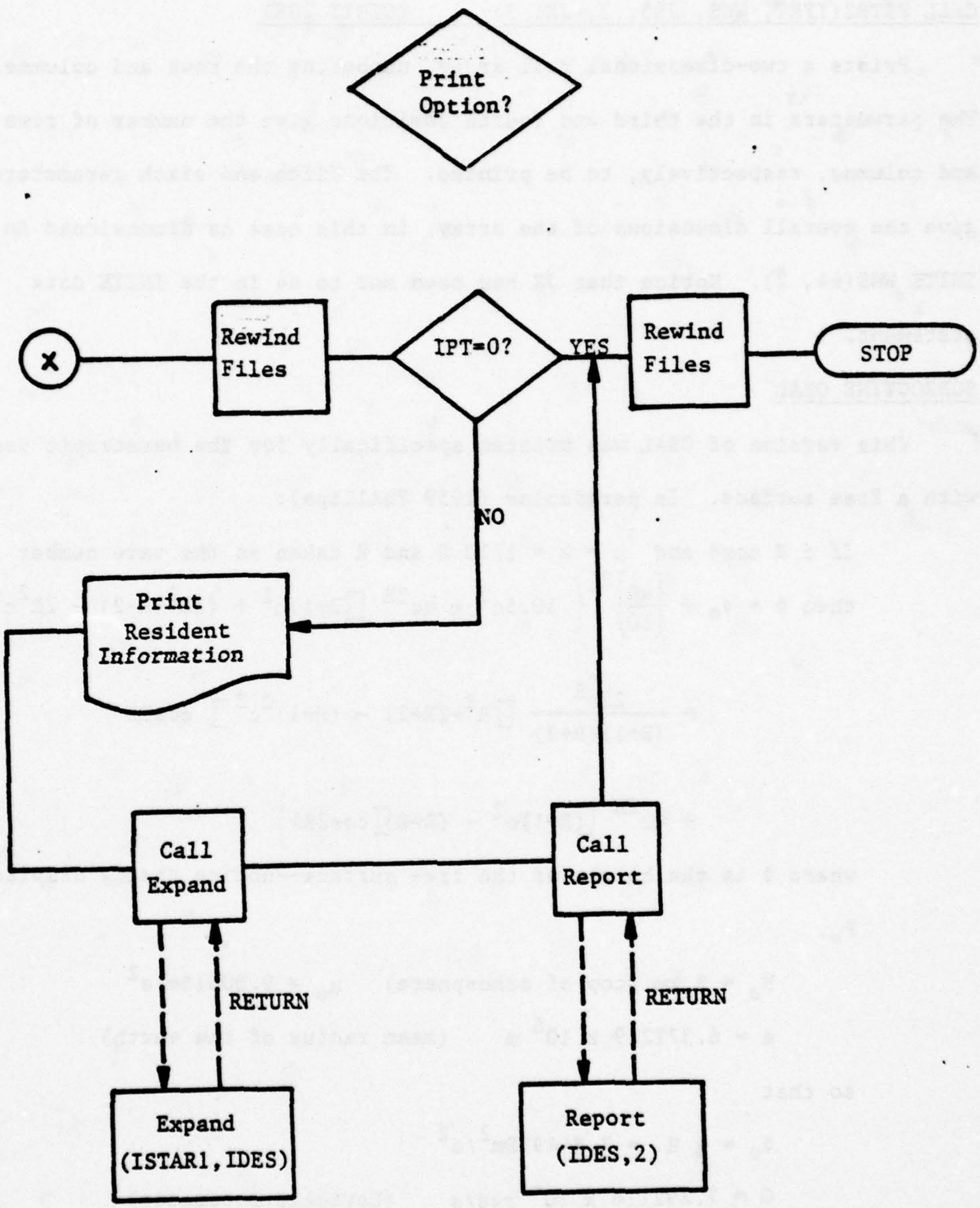
I1 = INIT $ UN = UNIT(I1) $ if (UN.GE.0) CALL DLAY (I1, 15, 1, UN)

If the unit function emerges not equal to -1, then there has been an error committed on the INIT file. The location of this call to DLAY can be decoded from the middle two parameters of the call. Within each subroutine, the calls to DLAY within that subroutine are given unique codes using these two parameters. DLAY processes the error by printing on the output file the unit on which the error occurred (here INIT), the location (15), and INDEX(1) of the call; and the nature of the error $\left\{\begin{array}{ll} 0 & \text{End of file read} \\ 1 & \text{Parity error} \end{array}\right\}$. See table of DLAY codes.

BUFFER OUT(I1,1) (CBLK(1), CBLK(1200))   (GINIT.170)

All the information in AC block (which has been equivalenced to CBLK) is output to the INIT file. After the placement of an EOF, the INIT file is ready.


Generate
Data.
Call
OBAL
Endfile
ISTAR1
#
x
Subroutine
OBAL
Buffer
out
3 word
label
ISTAR1
Fill
BLK
with
BLOCK L
DO
L=1,NBLK
Buffer
out
the
Lth Block
ISTAR1


Print Option?
X
Rewind Files
IPT=0?
YES
Rewind Files
STOP
NO
Print Resident Information
Call Expand
Call Report
RETURN
RETURN
Expand (ISTAR1, IDES)
Report (IDES,2)

Helpful Comments

CALL PRTR2(IPRT, WNS, JB5, 2, JX, 2) (GINIT.208)

Prints a two-dimensional real array, numbering the rows and columns. The parameters in the third and fourth positions give the number of rows and columns, respectively, to be printed. The fifth and sixth parameters give the overall dimensions of the array, in this case as dimensioned in INITE WNS(64, 2). Notice that JX has been set to 64 in the INITE data statement.

SUBROUTINE OBAL

This version of OBAL was written specifically for the baratropic case with a free surface. In particular (1959 Phillips):

If $c \equiv \cos\theta$ and $\omega = k = 1/10\ \Omega$ and R taken as the wave number

$$\text{then } \Phi = \Phi_o + \left(\frac{a\Omega}{10}\right)^2 \left\{ 10.5c^2 + \tfrac{1}{4}c^{2R}\left[(R+1)c^2 + (2R^2-R-2) - 2R^2c^{-2}\right]\right.$$

$$+ \frac{22c^R}{(R+1)(R+2)}\left[(R^2+2R+2) - (R+1)^2c^2\right]\cos R\lambda$$

$$\left. + \tfrac{1}{4}c^{2R}\left[(R+1)c^2 - (R+2)\right]\cos 2R\lambda\right\}$$

where Φ is the height of the free surface--notice that Φ displaces P_*.

H_o = 8 km (top of atmosphere) $g_o = 9.80616 \text{m/s}^2$

$a = 6.371229 \times 10^6$ m (mean radius of the earth)

so that

$\Phi_o = g H = 7.844928 \text{m}^2/\text{s}^2$

$\Omega = 7.292116 \times 10^{-5}$ rad/s (Coriolis parameter)

then $u = -\frac{1}{a}\frac{\partial\psi}{\partial\phi}$ $v = \frac{1}{a\cos\Theta}\frac{\partial\psi}{\partial\lambda}$

where

$$\psi = \frac{a^2\Omega}{10}\left\{-\sin\theta + c^R\sin\theta\cos R\lambda\right\}$$

so that

$$u = -\frac{a\Omega}{10}\left[-\cos\theta + \cos^{R-1}_{\theta}\ (\cos^2\theta - R\sin^2\theta)\cos R\lambda\right].$$

$$v = -\frac{a\Omega R}{10}\cos^{R-1}_{\theta}\ \sin\theta\ \sin R\lambda$$

Description of Variables

AL	$\alpha j + (i-1)\Delta\lambda j = \lambda$
ARE2	R^2
CRL	$\cos(R\lambda)$
CT	$\cos\theta j$
CTR	$\cos\theta j\ (\cos\theta j)^{R-1}$
CTRM	$(\cos\theta j)^{R-1}$
CT2	$\cos^2\theta j$
CTR2	$\cos^{2R}_{\theta j}$
C1	$a\Omega/10$
C12	$\left(\frac{a\Omega}{10}\right)^2$
C2	$\left(\frac{a\Omega}{10}\right)R$
DSJ	Proportional to ΔV of KH.
DTHJ	$\Delta\theta j = 2\alpha$
DXJ	$a\cos\theta j\Delta\lambda j$
DYJ	$a\ \Delta\theta j$
DZJ	$\Delta\sigma_k = \sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}$
I	Zonal index.
J	Meridional index.
J1	Loop.

J2	Loop.
K	Sigma index.
NBJ	Temporary variable holds number of cells in band j.
NT	Main loop index.
P1	$2R^2 - R - 2$
P2	$R^2 + 2R + 2$
P3	$(R+1)^2$
RP1	$R + 1$
RP1RP2	$(R+1)(R+2)$
RP2	$R + 2$
SRL	$\sin(R\lambda)$
ST	$\sin\theta_j$
THJ	Temporary variable to hold θ_j.
TR2	$2R^2$
UN	Returned value of unit function.
VEL	Magnitude of velocity vector.

<u>Helpful Comments</u>

EQUIVALENCE ((ISER, BLK(1), (ITS, BLK(2))

<u>ISER = NSER $ ITS = NTS $ BLK(3) = ADT (GOBALφ.13)</u>

Use of the equivalence avoids BLK(1) = NSER, which would initate a conversion to floating point. This 3 word identifying label appears as the first record of the data file.

<u>K1 = K2 $ K2 = K1 + 1 $ VMAX = 0 $ KMAX = IMAX = JMAX = KMIN = JMIN = 0</u>

These values will emerge as the indices of the locations of the maximum advecting velocity in the initial data, and the minimum space increment in the grid.

```
J2 = 1 $ DO 10 NT = 1, NBLK $ J1 = J1 + 1 $ JLI JLI(NT) $ LT = 0
DO 1 J = J1, J2                              (GOBALϕ.24)
```

The descriptive variables for the data file are here used to organize that file into NBLK data records, each record containing bands J1 thru J2.

The logic of the loops is:

```
NT = 1, NBLK
LT = 0
J = J1, J2
I = 1, NBJ
2-D variables P*, ϕ* for index (I,J) into BLK.
K = 1, KV
3-D variables u, v, T fill BLK vertically up column indexed (I,J)
5
4
1
BLK(LT)──────────────────────────> ISTAR1
Reinitialize LT = 0
10
```

```
UN = UNIT(ISTAR1) $ if (UN.GE.O) CALL DLAY (ISTAR1, 20, NT, UN)
```

Note that the second parameter (20), in the call to DLAY, locates the origin of this call uniquely within the outermost DO loop, and the third parameter (NT), which is the control index of the loop, will determine on which pass through the loop the error occurred.

```
Buffer out (ISTAR1, 1) (BLK( ), BLK( )) $ UN = UNIT(ISTAR1) $ LT = 0
```

The underlined statement has a history. The program had been run for a time with a 16 x 32 grid, and 3 prognostic variables. Since in this case there are (2 + 3 x 1) x 32 x 16 = 2560 data words in the initial data file, and the I/O size was 6528, only one block was required on the ISTAR1 file, and the loop indexed NT was passed through but once. However, when a finer grid, (32 x 64) was tried, bizarre results were obtained. A 32 x 64 grid requires (2 + 3 x 1) x 64 x 32 = 10240 data words, and, with an I/O size of 6528, this will require 2 blocks of data (NBLK = 2). Thus

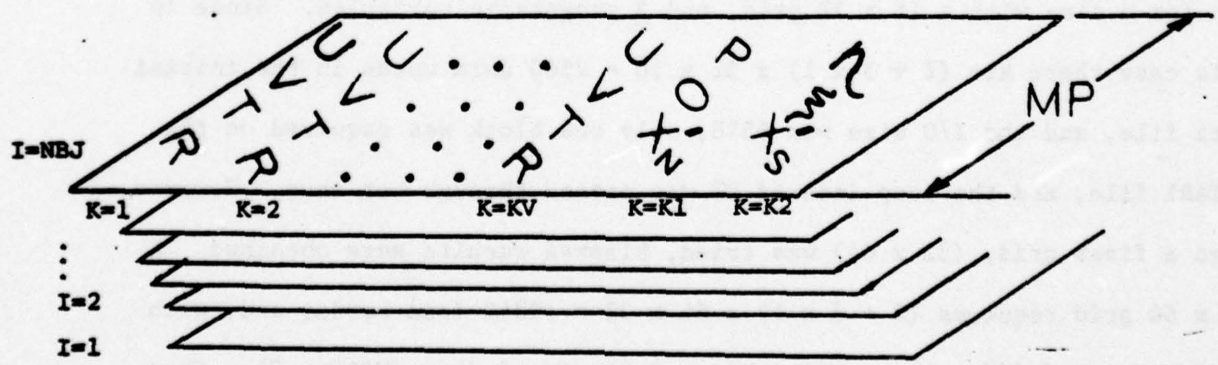
the outermost loop is passed through twice. Originally, however, the unit check was not performed at the end of the buffer out in GOBALϕ.64. Thus, new data was generated and placed on top of old data in BLK before the buffer was initialized, ie. the old data was destroyed before it was output. Unit checks will not be made at certain crucial points in GEX, to more efficiently overlap computation with I/O. But, a lesson here is that one must exercise care in attempts to increase efficiency this way.

EXPAND

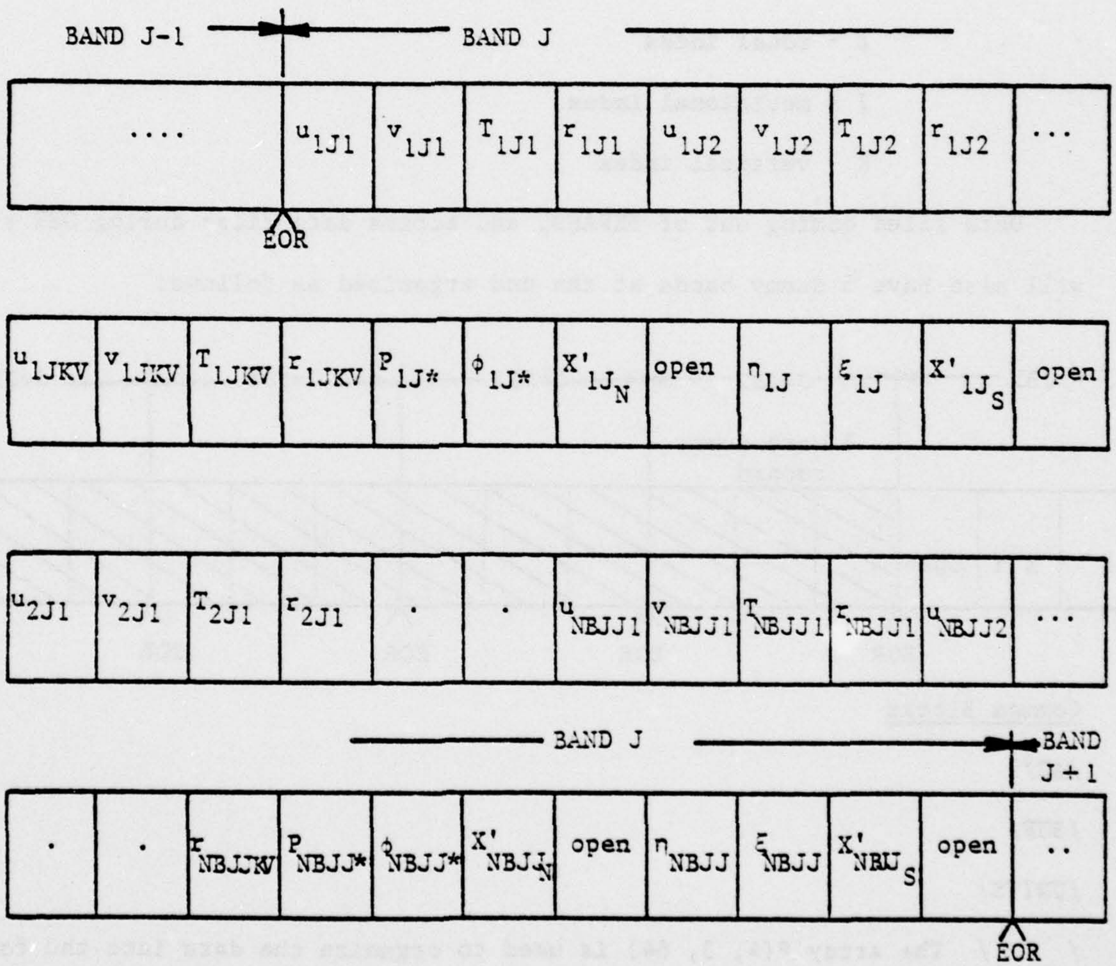
SUBROUTINE COMP(IU1, IU2)

ENTRY EXPAND

EXPAND is called from INITE with the formal parameters (ISTAR1, IDES) if the print option is in effect. In this case, it is necessary to reformat the ISTAR1 file (it's in compressed format), to prepare it for acceptance by REPORT. REPORT expects data for a time step NTS to be organized into JB1 records, plus 4 dummy records. The first record is a 3 word label, and subsequent records are full bands of data, output in arrays P(MP, K2, NBJ) (See KMII p. 7). That is, the data for band J is output as:



Since the CDC-6600 stores the 3-dimensional array as sequential planes $P(M,K)_I$, and stores each plane as a 1-dimensional array with the first subscript varying fastest, the actual storage for band J is :



where

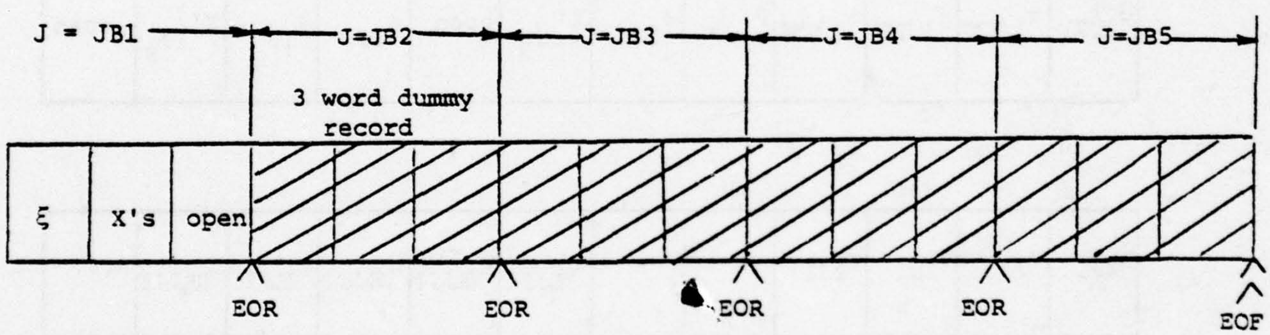
3-D Variable	2-D Variable	Grid Parameters	
u_{IJK}	P_{IJ*}	X'_{IJN}	η_{IJ}
v_{IJK}	ϕ_{IJ*}	X'_{IJS}	ξ_{IJ}
T_{IJK}			
r_{IJK}			

I · zonal index

J · meridional index

K · vertical index

Data filed coming out of EXPAND, and active data files during GEX runs, will also have 5 dummy bands at the end organized as follows:



<u>Common Blocks</u>

/AC/

/BUF/

/UNITS/

/ / The array P(4, 3, 64) is used to organize the data into the format discussed above.

Description of Variables

CLK Three word array used to hold label when compress entry is called. In this case a serial number check is performed before the file is output as a display file.

FI Zonal index (real). (See KMI p. 32)

I Zonal index (integer).

ITA1 $\eta_{1,i,j}$ (KMI)

ITA2 $\eta_{2,i,j}$ (KMI)

IXEN $\left[(X_N)_{b_j}\right]^{INT} = \left[b_{j-1} + (b_{j-1}/b_j)(\alpha_j - .5) + b_{j-1} - \alpha_{j-1} + .5\right]^{INT}$

IXES $\left[(X_S)_{b_j}\right]^{INT} = \left[b_{j+1} + (b_{j+1}/b_j)(\alpha_j - .5) + b_{j+1} - \alpha_{j+1} + .5\right]^{INT}$

IXI1 $\xi_{1,i,j}$

IXI2 $\xi_{2,i,j}$

J Meridional index.

J1 Control on loop expanding data block NT into P arrays.

J2 Control on loop expanding data block NT into P arrays.

K Level index.

KT Loop index for buffer on dummy bands.

L Index of JLI array when compressing data.

LT Index of array BLK into which data is being compressed, or out of which data is being expanded.

LZ Holds size of successive data blocks in compressed file.

M Index for 3-dimensional variables.

NBJ b_j

NBM1 b_{j-1}

NBP1 b_{j+1}

NHOLD Equivalent to CLK(2). Avoids ITS = CLK(2) in compress.

NT Index for loop on compressed data blocks.

NUN ν_N (KMI)

NUS ν_S (KMI)

RB1 b_{j-1}/b_j

RB2 b_{j+1}/b_j

TA1 $b_{j-1}/b_j\ (\alpha_{j-.5}) + b_{j-1} - \alpha_{j-1} + .5$

TA2 $b_{j+1}/b_j\ (\alpha_{j-.5}) + b_{j+1} - \alpha_{j+1} + .5$

UN Returned value of unit function.

WN $W_{\Delta\lambda_j}{}^{N}_{j-1}$

WS $W_{\Delta\lambda_j}{}^{S}_{j+1}$

XN $i(b_{j-1}/b_j) + (b_{j-1}/b_j)(\alpha_{j-.5}) + b_{j-1} - \alpha_{j-1} + .5$

XS $i(b_{j+1}/b_j) + (b_{j+1}/b_j)(\alpha_{j-.5}) + b_{j+1} - \alpha_{j+1} + .5$

Helpful Comments

ENTRY EXPAND (GCE.32)

Calls to COMP will be analyzed in the natural sequence they will occur.

C EXPAND FILE ON IU1 BUFFER TO IU2 (GCE.33)

Note that the file is assumed positioned by the calling program.

NTS = ITS \$ ADT = BLK(3) \$ NSER = ISER (GCE.37)

Save 3 word label off this data file. In a GEX run, with restart data in two time steps ISTAR1 and ISTAR2, the first file expanded is step N-1, the next is step N. In that case the statement above leaves NTS = N on return to GEX (as it should!), and the next time step prognosticated will be N+1.

J2 = 1 \$ DO 200 NT = 1, NBLK \$ LT = 0
LZ = LSZ(NT)

The descriptive variables for the compressed file are here used to organize that file into an expanded file, of 1 label record, followed by

JB data records, followed by 4 dummy records. The LSZ array carries the size of each of the NBLK compressed data blocks.

The loop begins with the compressed data file positioned after the label, and before the first data block.

The logic of the loops is:

```
  ┌── NT = 1, NBLK
  │   LT = 0
  │   LZ = LSZ(NT)
  │  Thru BLK(LZ) <──────────────────  COMPRESSED FILE IU1
  ├── J = J1, J2
  │   Define parameters for this band J
  │ ┌─ I = 1, NBJ
  │ │ Define X_N, X_S, η, ζ, ν_N, ν_S for this cell I
  │ │ P(1,K2,I) = η_2·ν_N      P(3,K1,I) = X_N'
  │ │ P(2,K2,I) = ξ_2·ν_S      P(3,K2,I) = X_S'
  │ │ P(1,K1,I) = P_*
  │ │ P(2,K1,I) = φ_*
  │ ├─ K = 1,KV
  │ ├─ M = 1,MP
  │ │ P(M,K,I) = 3-D prognostic variable u, v, T or r
  │ └─ 150
  │   P(MX,K2,NBJ) ──────────────────>  EXPANDED FILE IU2
  └── 200
  ┌── KT = 1,4
  │   BLK(3) ────────────────────────>  EXPANDED FILE IU2
  └── 13
```

Note that the size of the data records buffered out is flexible. MX, K2 are fixed in P(MX,K2,NBJ), but, NBJ may vary band by band. MX is used rather than MP, since the prognostic variables occupy fixed positions within each planar segment of P.

TA1 = RB1 * (DEV(J)-.5) + FNB(J-1) - DEV(J-1) + .5
TA2 = RB2 * (DEV(J)-.5) + FNB(J+1) - DEV(J+1) + .5 (GCE.51)

That is $TA1 = (\alpha_{j-.5})\, b_{j-1}/b_j + b_{j-1} - \alpha_{j-1} + .5$

$TA2 = (\alpha_{j-.5})\, b_{j+1}/b_j + b_{j+1} - \alpha_{j+1} + .5$

IXEN = FNB(J) * RB1 + TA1 $ IXES = FNB(J) * RB2 + TA2 (GCE.52)

That is $IXEN = b_{j-1} + (\alpha_{j-.5})\, b_{j-1}/b_j + b_{j-1} - \alpha_{j-1} + .5$

$IXES = b_{j+1} + (\alpha_{j-.5})\, b_{j+1}/b_j + b_{j+1} - \alpha_{j+1} + .5$

Notice the conversion to integer.

Now $X_N = i(b_{j-1}/b_j) + [(\alpha_{j-.5})(b_{j-1}/b_j) + (b_{j-1}-\alpha_{j-1} + .5)]$

$X_S = i(b_{j+1}/b_j) + [(\alpha_{j-.5})(b_{j+1}/b_j) + (b_{j+1}-\alpha_{j+1} + .5)]$

(See KMI 3.5.3).

$$\left.\begin{aligned} \text{Therefore IXEN} &= [X_N]^{INT} \\ \text{IXES} &= [X_S]^{INT} \end{aligned}\right\} \text{for I} = b_j$$

That is, these values are initialized to that corresponding to the last cell in the band j with index b_j, <u>before</u> entering the i loop.

<u>ITA2 = MOD(IXEN,NBM1) + 1 $ IXI2 = MOD(IXES,NBP1) + 1</u>

That is $\eta'_{2,i,j} = \left[[X_N]^{INT} \right]^{MOD(b_j-1)} + 1.$

$\xi'_{2,i,j} = \left[[X_S]^{INT} \right]^{MOD(b_j+1)} + 1.$ (KMI 3.5.4)

<u>FI = I $ XN = FI * RB1 + TA1 $ XS = FI * RB2 + TA2</u>

These are the forms for X_N, X_S given above for cell i.

<u>P(3,K1,I) = WN * AMOD(XN,1.) $ P(3,K2,I) = WS * AMOD(XS,1.)</u>

That is $P(3,K1,I) = X'_N = W_{\Delta\lambda^N_{2,i,j}} = W_{\Delta\lambda^N_{j-1}} [X_N]^{MOD(1)}$

$P(3,K2,I) = X'_S = W_{\Delta\lambda^S_{j+1}} [X_S]^{MOD(1)}$

(See KMI 3.72)

<u>ITA1 = ITA2 $ IXI1 = IX12 (GCE.57)</u>

That is $\eta'_{1,i,j} = \eta'_{2,i-1,j}$

$\xi'_{1,i,j} = \xi'_{2,i-1,j}$

(See KMI 3.3.11).

On the first pass through the loop, for example, this statement sets the left northern boundary index of the first cell on the band $\eta'_{1,1,j}$, equal to the right northern boundary index of the last cell on the band $\eta'_{2,b_j-1,j}$.

This value has been preset (at GCE.53) before the i loop was entered.

ITA2 = MOD(INT(X_N),NBM1) + 1 $ IX12 = MOD(INT(X_S),NBP1) + 1

This time the more general values of X_N, X_S for the cell index i are used.

CALL NU(ITA1, ITA2, NBM1 NUN) $ CALL NU(IXI1 IXI2, NBP1, NUS)

Returns the coded value of ν for north boundary ν_N, and the south boundary ν_S, for cell (i,j). $1 \leq \nu \leq 6$.

P(1,K2,I) = FLOAT(ITA2) + .1 * FLOAT(NUN)
P(2,K2,I) = FLOAT(IXI2) + .1 * FLOAT(NUS) (GCE.61)

Maximum compression of data. Notice, for example, that only the right hand cell index, $\eta'_{2,i,j}$, is carried since the left hand index, $\eta'_{1,i,j}$, can be recovered from η_2 of the previous contiguous cell (i-1,j), ie. $\eta'_{1,i,j} = \eta'_{2,i-1,j}$.

The coded value of ν is packed in the decimal part of the word, the values of η, ξ in the integer part.

150 P(M,K,I) = BLK(LT) (GCE.64)

Data as input off the display file fleshes out the P array.

6 BUFFER OUT (1U2,1) (P(1,1,1), P(MX,K2,NBJ) (GCE.66)

Note again the calls to DLAY are uniquely identified, within the subroutine, by the second parameter, and, within the loop, by the third parameter.

BUFFER OUT (1U2,1) (BLK(1),BLK(3)) (GCE.71)

Four 3 word dummy records corresponding to bands indexed JB2, JB3, JB4 JB5, and used to implement some of the time integration schemes.

ITS = NTS $ BLK(3) = ADT $ ISER = NSER (GCE.74)

Returns label to first 3 words of BLK before leaving routine (REPORT expects to find these values there for example).

SUBROUTINE NU(I1,I2,NB,NUT)

This routine returns the coded value of ν_N, ν_S for the cell i on the band j. Notice that since the grid if organized in non-overlapping vertical columns, these values will serve for all k levels.

Description of Variables

I1 Westernmost contiguous cell
For north boundary $(\eta'_{1,i,j,j-1})$

For south boundary $(\xi'_{1,i,j,j+1})$

I2 Easternmost contiguous cell
For north boundary $(\eta'_{2,i,j,j-1})$

For south boundary $(\xi'_{2,i,j,j+1})$

NB Number of cells on contiguous band
For north boundary b_{j-1}
For south boundary b_{j+1}

NUM Temporary variable to hold coded value of ν.

NUT Coded value of ν. See description ahead.

NU1 $\eta_2 - (\eta_1+1)$ or $\xi_2 - (\xi_1+1)$

NU2 $b_{j\pm 1} + \eta_2 - (\eta_1+1)$ or $b_{j\pm 1} + \xi_2 - (\xi_1+1)$

NUT

	Value NUT	
$\nu \geq -1$	1	$\nu = 0$ Easternmost and westernmost cells are contiguous
	2	$\nu = -1$ Eastermost and westernmost cells coincide
	3	$\nu = \eta_2 - \eta_1$ or $\xi_2 - \xi_1$
$\nu < -1$	4	$b_{j-1} + \eta_2 - (\eta_1+1) > 0$ $(b_{j+1} + \xi_2 - (\xi_1+1) > 0)$ $\eta_1 \geq b_{j-1}$ $(\xi_1 \geq b_{j+1})$
	5	$b_{j-1} + \eta_2 - (\eta_1+1) \geq 0$ $(b_{j+1} + \xi_1 - (\xi_1+1) \geq 0)$ $\eta_1 < b_{j-1}$ $(\xi_1 < b_{j+1})$ $\eta_2 \leq 1$ $(\xi_2 \leq 1)$

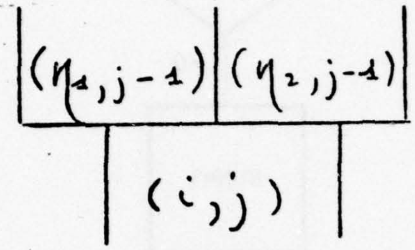
6 $\quad b_{j-1} + \eta_2 - (\eta_1+1) > 0 \quad (b_{j+1} + \xi_2 - (\xi_1+1) > 0)$

$\eta_1 < b_{j-1} \; (\xi_1 < b_{j+1})$

$\eta_2 > 1 \quad (\xi_2 > 1)$

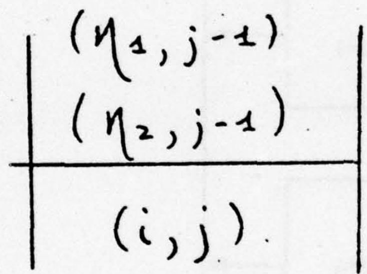
NUN = 1

For example



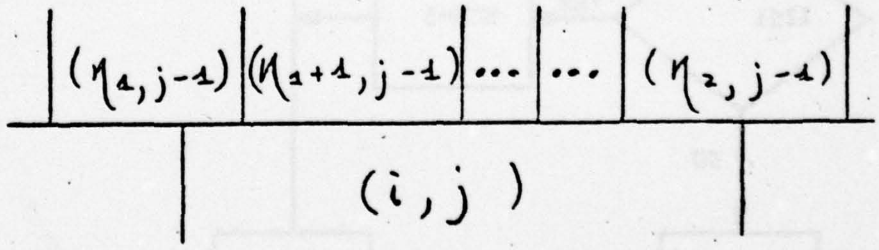
NUN = 2

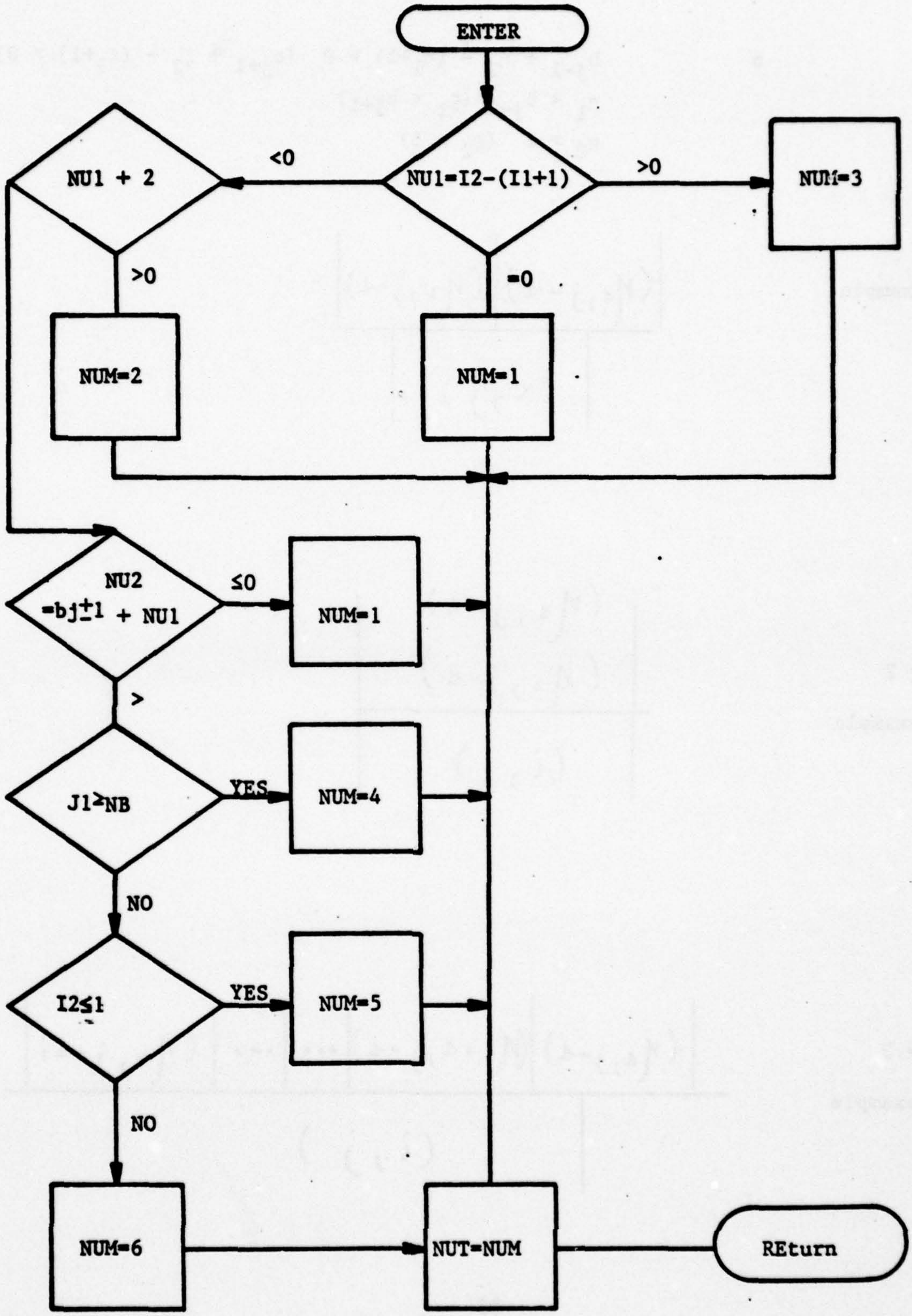
For example



NUN = 3

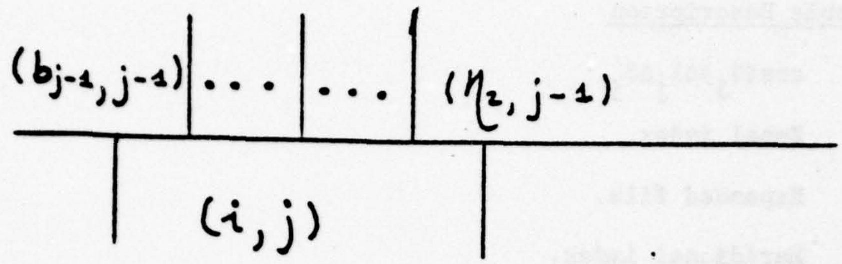
For example




ENTER
NU1=I2-(I1+1)
<0
>0
=0
NU1 + 2
≤0
>0
NUM=3
NUM=2
NUM=1
NU2
=bj±1 + NU1
≤0
NUM=1
>
J1≥NB
YES
NUM=4
NO
I2≤1
YES
NUM=5
NO
NUM=6
NUT=NUM
REturn

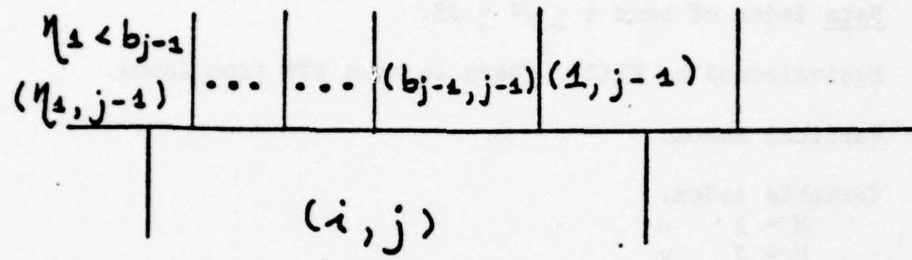
NUN = 4

For example



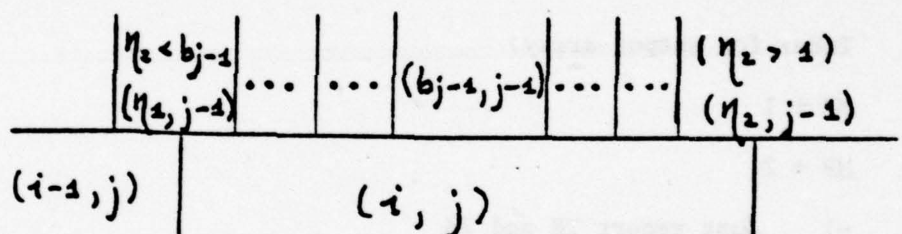
NUN = 5

For example



NUN = 6

For example



<u>SUBROUTINE REPORT (IU, MR)</u>

Selected data from file IU, which has just been expanded, is reported on the community output file. REPORT is called from DISPLAY with MR = 1 or MR = -1, and from INITE with MR = 2.

<u>Common Blocks</u>

/BUF/

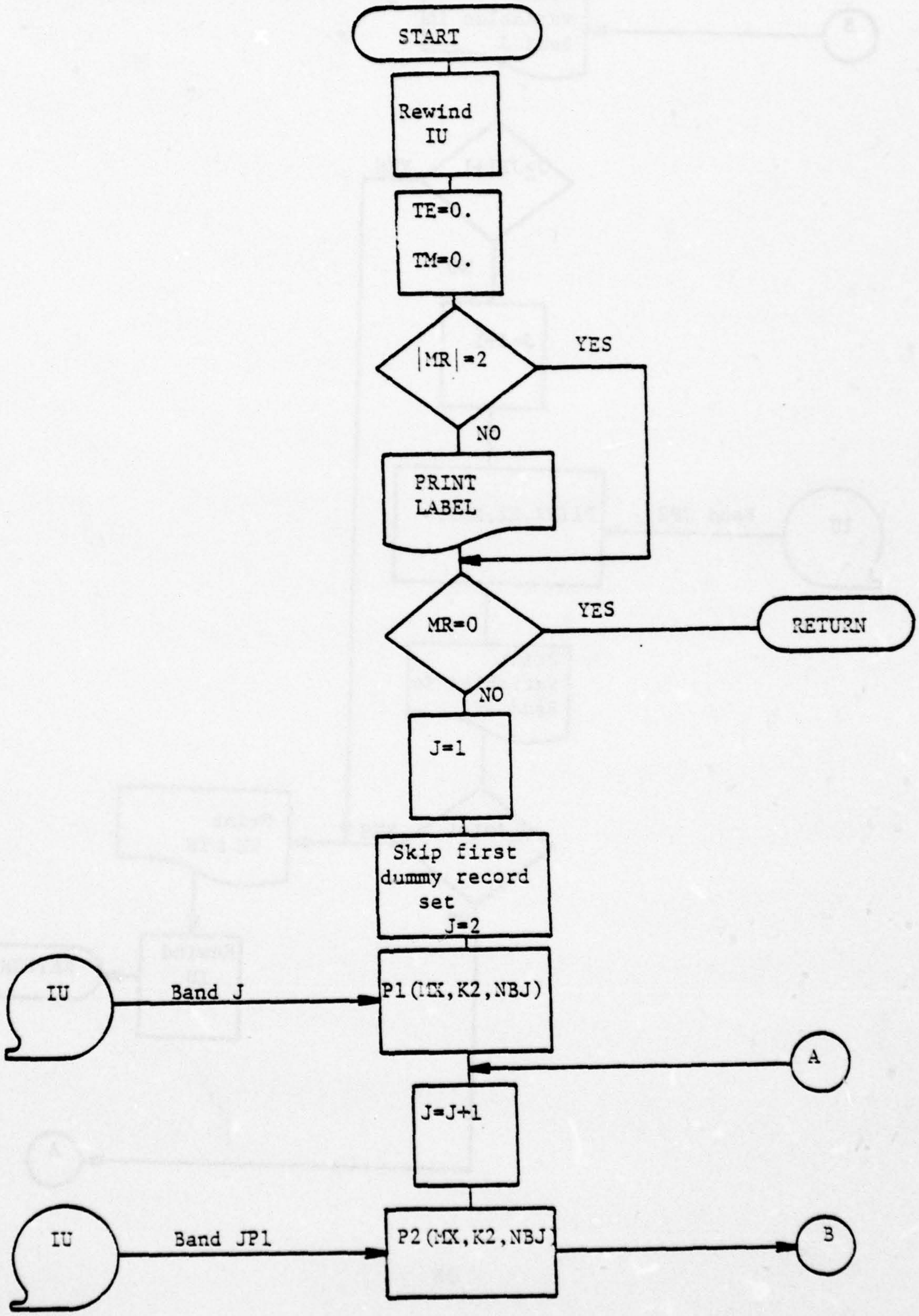
/UNITS/

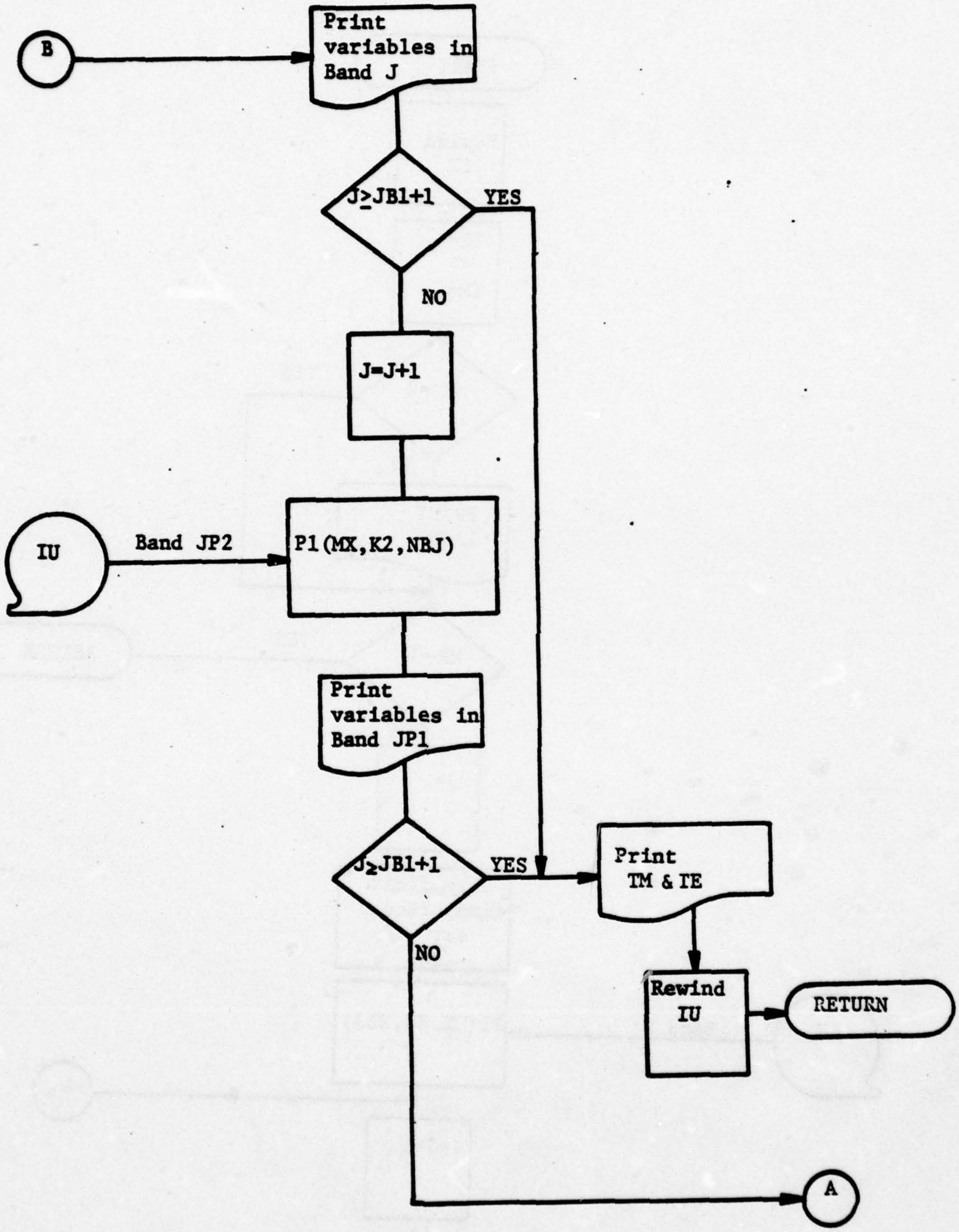
/AC/

/ /

Variable Description

CT	$\cos(\theta_j)\Delta\lambda_j\Delta\theta_j$
I	Zonal index.
IU	Expanded file.
J	Meridional ixdex.
JB3	JB1 + 1
JM	J - 1. Current value of J being processed.
JP	Data index of band 1 $\leq$ JP $\leq$ JB.
JTS	Equivalenced to P1(2). Used to hold NTS from label.
K	Vertical index.
M	Variable index. M = 1 u M = 2 v M = 3 T M = 4 r
MM	Index for output array.
MP1	MP + 1
MP2	MP + 2
MR	-1 Just report TE and TM 0 Just report label and real time clock 1 Called from DISPLY 2 Called from INITE
NBJ	b_j
PE	Partial energy. Energy in band j.
PM	Partial mass. Mass in band j.
TE	Total energy.
TM	Total mass.


START
Rewind IU
TE=0.
TM=0.
|MR|=2
YES
NO
PRINT LABEL
MR=0
YES
RETURN
NO
J=1
Skip first dummy record set
J=2
IU
Band J
P1(MX,K2,NBJ)
A
J=J+1
IU
Band JP1
P2(MX,K2,NBJ)
B


B
Print variables in Band J
J≥JB1+1
YES
NO
J=J+1
IU
Band JP2
P1(MX,K2,NBJ)
Print variables in Band JP1
J≥JB1+1
YES
NO
Print TM & TE
Rewind IU
RETURN
A

The logic of the loops is:

```
K = 1, KV
PE = 0, PM = 0
I = 1, NBJ
M = 1, MP
P3(M) = P1(M,K,I)          3-D prognostic variables
                           for cell (i,j,k)
12
P3(MP1) = P1(1,K1,I)       P4 for (i,j,k)
P3(MP2) = P3(MP1) - 8000
PE = PE + P*(u^2 + v^2)
PM = PM + P*
PE = PE + g_o (P*)^2
IF (MR ≤ 0) GO TO 11
WRITE P3
11
PM = PM*cosθ_j Δλ_j Δθ_j
PE = PE*cosθ_j Δλ_j Δθ_j
WRITE PE, PM
10
```

Helpful Comments

EQUIVALENCE (P1(2), JTS)   (GRPORT.20)

JTS holds time step number after label is buffered into P1.

REWIND IU

After leaving expand the file IU is positioned at the end of the last dummy band JB5.

WRITE (6, 200) ITS, BLK(3), RTJ   (GRPORT.29)

Before leaving EXPAND, the label was returned to BLK(1)--BLK(3).

39 BUFFER IN (IU,1) (P1(1,1,1),P1(MX,K2,NB5))

Only the 3 word label is buffered.

J = 2
NBJ = NB(5)   (GRPORT.49)

J = 2 starts the first data band, with NBJ cells.

BUFFER IN (IU,1) (P2(1,1,1), P2(MX,K2,NBJ))   (GRPORT.56)

Notice that there is not a unit check, immediately after this buffer. In this way the data from the j^{th} band (P1), can be manipulated simultaneously

while the buffer for the J+1st band (P2) is being initialized.

JM = J - 1
NBJM = NB(JM) (GRPORT.59)

JM is now the meridional index of the band in P1.

PE = 0. $ PM = 0. (GRPORT.61)

Energies and masses are reported for each band at each fixed level. Here partial energy and mass are reinitialized before reporting the next level.

P3(M) = P1(M,K,I) (GRPORT.64)

P3 is used only to hold the selected variables to be printed.

P3(MP2) = P3(MP1) - 8000.

A convenience for scanning the output.

PE = PE + P3(MP1) * (P3(1)**2 + P3(2)**2) $ PM = PM + P3(MP1)
PE = PE + GO * P3(MP1)**2 (GRPORT.69)

Energy = Kinetic Energy + Potential Energy

$$= \sum_{i,j,k} P_* \{(u^2 + v^2) + g_0 P_*\}$$

In the baratropic case $P_* = H$ = height of free surface

Mass $= \sum_{i,j,k} P_*$ where $(P_*)_{i,j} = H_{i,j}$ - ΔMass above (i,j) (See 4.6 KH)

JP = JM - 1 (GRPORT.70)

JP is the index of the present band in the data bands ($1 \leq JP \leq JB$ whereas $1 \leq J \leq JB5$).

PM = PM * CT $ PE = PE * CT $ TE = TE + PE $ TM = TM + PM

$(\text{Mass})_{j,k} = (\Delta M)_{j,k}\, w_{ij}$

$(\text{Energy})_{j,k} = (\Delta E)_{j,k}\, w_{i,j}$

where $(\Delta M)_{j,k}$ is mass per unit area

$(\Delta E)_{j,k}$ is energy per unit area.

$w_{i,j} = \cos\theta_j \Delta\lambda_j \Delta\theta_j$ weights the cell areas for band J.

WRITE (IPRT, 700) NSER, NTS, JP, PE, PM

NTS was saved as the value of JTS. Partial energies and masses are reported after the prognostic variables for each band, and for each level.

IF (J - JB1) 15, 16, 16 (GRPORT.81)

When J = JB3 = JB1 + 1 = JB2 all data bands have been reported.

17 BUFFER IN (IU,1) (P1(1,1,1), P1(MX,K2,NBJ)) (GRPORT.85)

Notice again no unit check immediately after the buffer. The data from the J+1st band (P2) can be manipulated (simultaneously) while the buffer for the J+2nd band (P1) is being initialized.

JM = J - 1
NBJM = NB(J-1) (GRPORT.87)

JM is the meridional index of the band in P2.

APPENDIX B

GEX "Global Experiment"

GEX integrates the variables in time and saves user specified time steps for later display.

Characteristics

Core: 70000B

Time: Open

Options: Initial data or restart data
Choice of time integration schemes
Choice of time steps to be saved

Permanent Files: Display file

Organization of Core During a Run

Named Common	/BUF/	101	14600
	/UNITS/	14701	15
	/AC/	14716	2260
Programs	GEX	17176	4766
	REPORT		
	/FM/		
	/FRC/		
	/LTC/		
	/CEH/		
	STISϕ		
	/FG/		
	/BC/		
	/CEF/		
	CBUFN	35632	212
SYSTEM			
Blank Common	/ /	54746	12000

Common Blocks

/AC/ As in INITE.

/BUF/ Used as buffer area for output (input) of display files.

/UNITS/ As in INITE.

/ / Each PJ array holds 1 latitude band in expanded format. There are a maximum of 12 bands needed in core (for TIS Euler trapezoidal).

Files

INPUT(5)--Community Input File

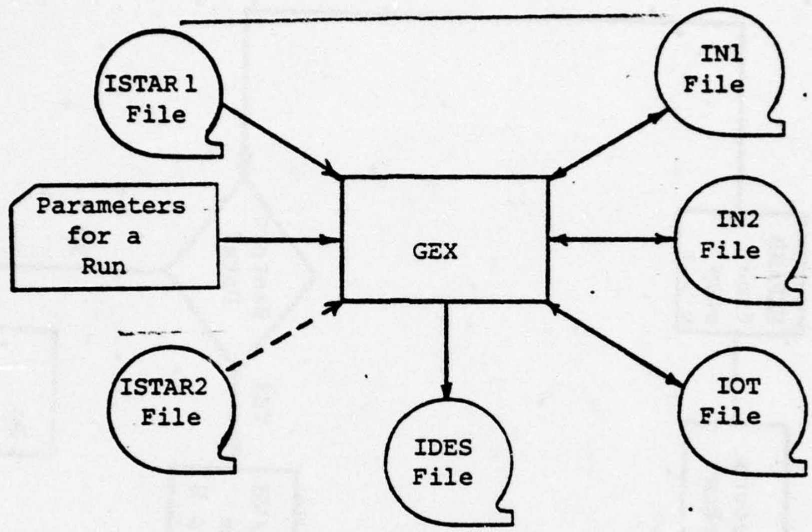
OUTPUT(6)--Community Output File

INIT(7)--Private Binary File of Grid Information

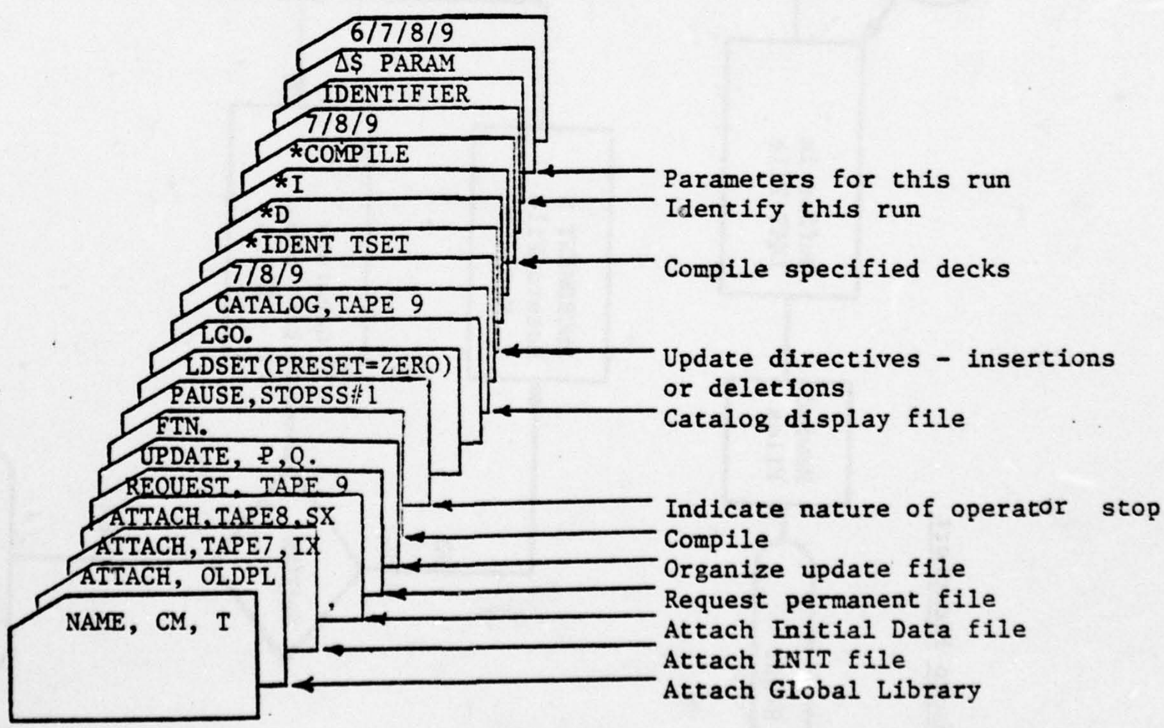
ISTAR1(8)--Initial data file created in OBAL (ts = 0) or restart data file (ts = N - 1)

ISTAR2(0, 8, or other)--Not used (0) or restart data file sequentially following on ISTAR1 file (8), or found on another

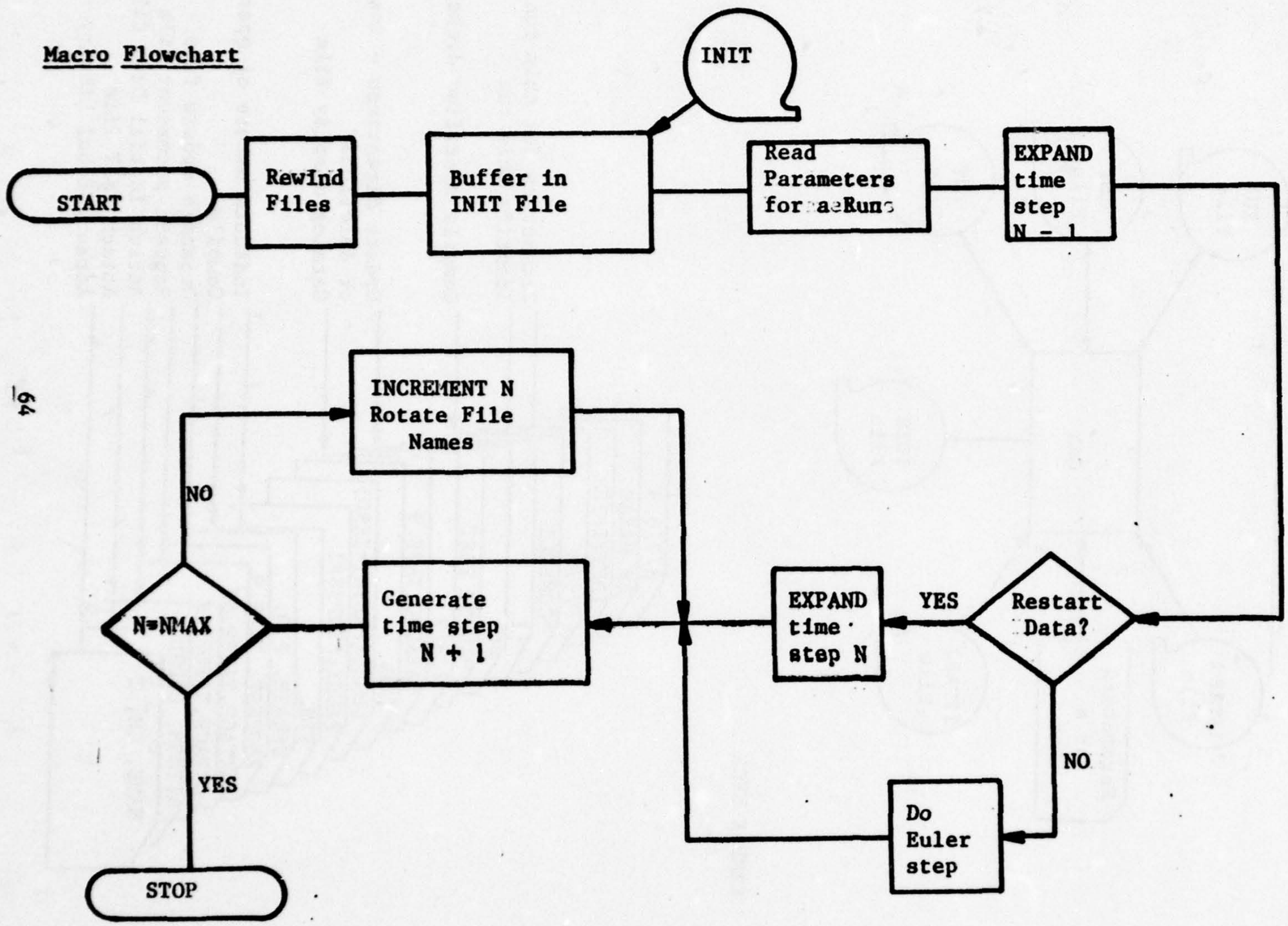
INPUT OUTPUT CHART



SAMPLE DECK



Macro Flowchart



file (ts = N).

IDES(9)--Display file of compressed data stored as a permanent file

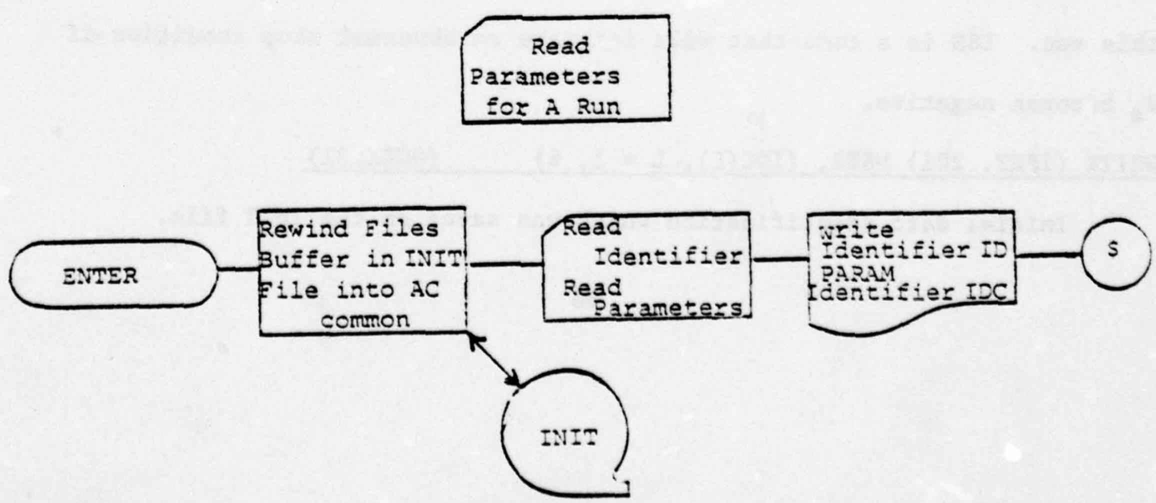
IN1(10, 11, 12)--Active data file holding ts N.

IN2(10, 11, 12)--Active data file holding ts N-1.

IOT(10, 11, 12)--Active data file holding ts N+1.

Description of Variables

FTI	Fixed time of integration. Read in on PARAM card, it determines the value of Δt(DT) for this run.
ID	70 character identification for this run.
IDDY	Holds value of IN1 during process of rotating file names.
IDY	Holds value of IN1 during process of interchanging file names IN1 and IN2 holding restart data.
INPUT	Community input file.
IN1	Active data file.
IN2	Active data file.
IRE	Nor presently used.
L	Index for output of identifier array ID.
U	Returned value of unit function.
UN	Returned value of unit function.



Helpful Comments

REWIND INIT
REWIND IOT $ REWIND IN1 $ REWIND IN2 (GGEX.16)

For restart data, ISTAR1 (ISTAR2) files are positioned by SCOPE cards before the two consecutive time steps for which the integration will be continued. ISTAR1, ISTAR2 files should not be rewound.

BUFFER IN (INIT,1) (CBLK(1), CBLK(1200)) (GGEX.18)

AU of the information describing the grid, and the arrangement of data storage is now in AC block. This information will be accessed for individual cells in TIS.

READ (INPUT, PARAM) (GGEX.22)

Required here is at least FTI, MTS, DTI, RTL, IDESF.

DT = FTI
DTH = .5 * DT
DTD = 2. * DT (GGEX.28)

Δt (Euler), $\frac{\Delta t}{2}$ (Adams-Bashforth), $2\Delta t$ (Leapfrog)

ZB(1) = 0. $ ZB(KV1) = 0.

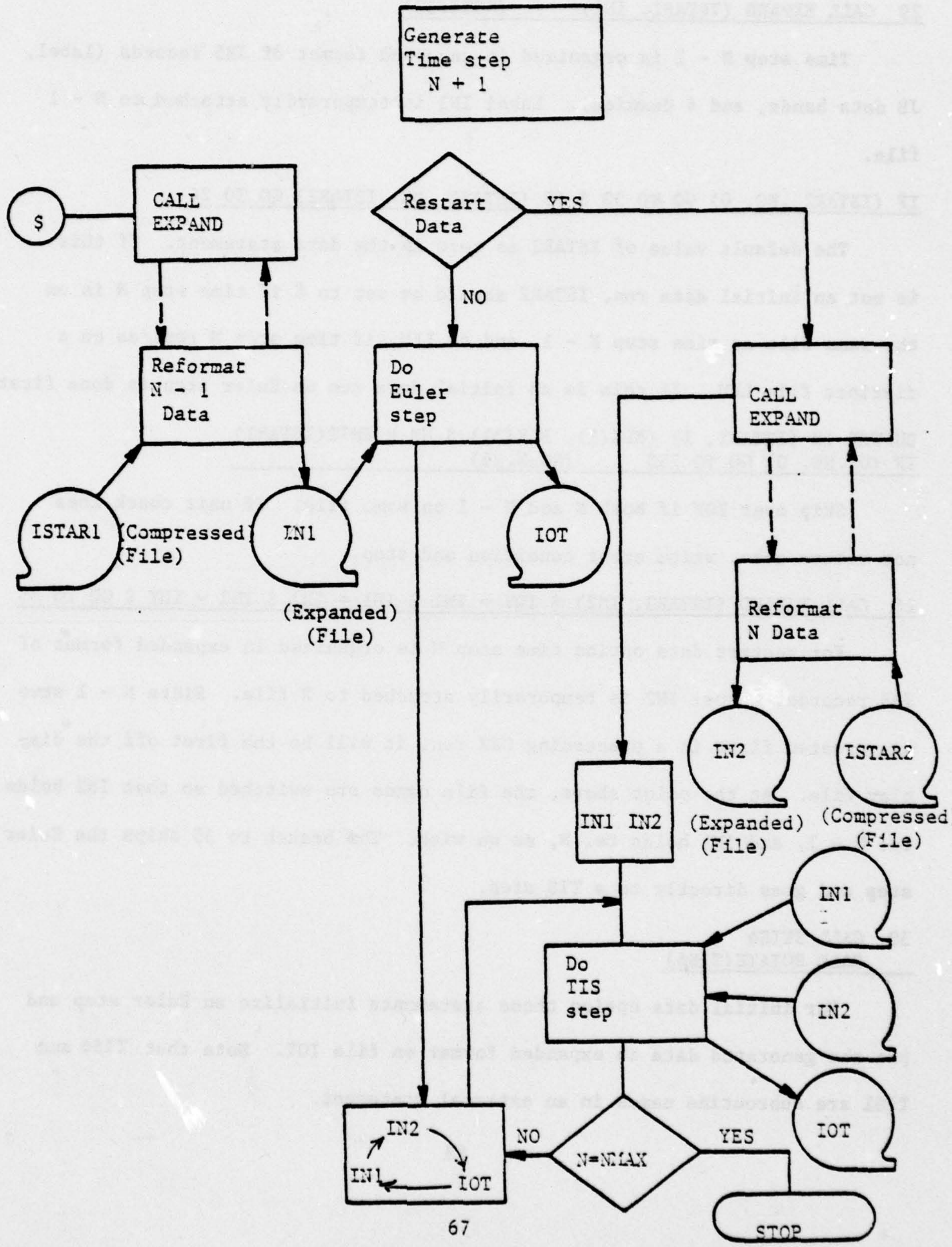
$\overline{\omega} = 0$ at the top ($k = 1$), and bottom (KV1) of the atmosphere.

NDES = ISS = 0

NDES will keep a running count of how many display files are output in this run. ISS is a code that will initiate an abnormal stop condition if P_* becomes negative.

WRITE (IPRT, 201) NSER, (IDC(L), L = 1, 6) (GGEX.31)

Initial data identification which was saved on the INIT file.


Generate
Time step
N + 1
Restart
Data
YES
NO
$
CALL
EXPAND
Reformat
N - 1
Data
Do
Euler
step
CALL
EXPAND
ISTAR1
(Compressed)
(File)
IN1
(Expanded)
(File)
IOT
Reformat
N Data
IN2
ISTAR2
(Expanded)
(File)
(Compressed)
(File)
IN1 IN2
IN1
Do
TIS
step
IN2
IOT
IN2
IN1
IOT
NO
N=NMAX
YES
STOP

Helpful Comments

29 CALL EXPAND (ISTAR1, IN1) (GGEX.33)

Time step N - 1 is organized in expanded format of JB5 records (label, JB data bands, and 4 dummies). Label IN1 is temporarily attached to N - 1 file.

IF (ISTAR2 .EQ. 0) GO TO 30 $ IF (ISTAR1 .NE. ISTAR2) GO TO 26

The default value of ISTAR2 is zero in the data statement. If this is not an initial data run, ISTAR2 should be set to 8 if time step N is on the same file as time step N - 1, and to LFN, if time step N resides on a distinct file LFN. If this is an initial data run an Euler step is done first.

BUFFER IN (ISTAR1, 1) (BLK(1), BLK(3)) $ UN = UNIT(ISTAR1)
IF (U .NE. 0) GO TO 750 (GGEX.36)

Skip over EOF if both N and N - 1 on same file. If unit check does not return zero, write error condition and stop.

26 CALL EXPAND (ISTAR2, IN2) $ IDY = IN1 $ IN1 = IN2 $ IN2 = IDY $ GO TO 35

For restart data option time step N is organized in expanded format of JB5 records. Label IN2 is temporarily attached to N file. Since N - 1 step was created first in a preceeding GEX run, it will be the first off the display file. At the point above, the file names are switched so that IN2 holds ts. N - 1, and IN1 holds ts. N, as we wish. The branch to 35 skips the Euler step and goes directly to a TIS step.

30 CALL STISϕ
CALL ROTATE(TISϕ)

For initial data option these statements initialize an Euler step and put the generated data in expanded format on file IOT. Note that TISϕ and TIS1 are subroutine names in an external statement.

34 IDDY = IN1 $ IN1 = IOT $ IOT = IN2 $ IN2 = IDDY

To avoid the actual transfer of data, the file names are rotated. In this way

(12)	IN2	N - 1
(11)	IN1	N
(10)	IOT	N + 1

becomes

(11)	IN2	N
(10)	IN1	N + 1
(12)	IOT	N - 1

and the stage is set for the creation of ts. N + 2 in IOT.

```
35 CALL STIS1
   CALL ROTATE(TIS1)
```

A TIS step is initialized and the generated data, in expanded format, is put on file IOT. The nature of the TIS (Leapfrog, Adams-Bashforth, Leapfrog Trapezoidal, etc.) is determined by the update * COMPILE card. All modules share the same subroutine names, but have unique deck names, enabling the user to compile a specific TIS from his Global library, and then use the above Fortran calls to execute that TIS.

IF (NTS .LE. MTS) GO TO 34

Note that the return to 34 rotates the file names, before pressing ahead with the creation of the next ts. in IOT.

SUBROUTINE STISϕ

Start time integration scheme Euler. This routine sets up data arrays in central memory in preparation for going into TIS at J = 2 (first data band).

Common Blocks

$\left.\begin{array}{l}/\ \ /\\ /BUF/\\ /UNITS/\\ /AC/\end{array}\right\}$ As previously described

/LTC/ Codes and other information, used in creating display files while time integration scheme is proceeding.

/CEF/ Grid information for a particular cell (i,j,k). This information is passed to the routines which calculate the tendencies for that cell (which are also held in this common block).

/CEH/ Holds b_j for a fixed band being processed.

/CEFG/ Holds the zonal index for a specific column of cells being processed.

Description of Variables

COP Coriolis Parameter $\hat{f} = 2\Omega\sin\theta_j \cos(\Delta\theta_j/2)$ for current band being processed.

COT Metric term in equation of motion $\hat{m} = \tan\theta_j/a$ for current band being processed.

D3 Integral of 3-dimensional divergence for the 4 prognostic variables at 25 possible levels.

I Holds zonal index for current cell column being processed by RHS.

ICE Code to locate origin of call to buffer error subroutine, within this subroutine. These codes are unique to each routine (See appendix for BERROR codes).

IDX Code to indicate if the time step currently being processed is a display file.

IDX = 0 Not a display file
IDX = 1 Display file

IM Zonal index of cell immediately west of current cell.

IOPS Code for sense switch # 1.

IOPS = 1 Operator stop indicated
ICPS = 2 Processing continues.

IP Zonal index of cell immediately east of current cell.

ITA Holds η_1, η_2 for current cell column--ITA(1) = η_1, ITA(2) = η_2.

IUN Identifies file on which buffer error occurred--passed to BERROR subroutine.

IXI Holds ξ_1, ξ_2 for current cell column--IXI(1) = ξ_1, IXI(2) = ξ_2.

JT Identifies call to BERROR routine as having originated in STIS subroutine (JT = 0).

LB Index for JLI array. If current ts. is a display file (IDX = 1), then LB is incremented each time a block of data goes out as a record on the display file.

LT Index for BLK array. If current ts. is a display file (IDX = 1), then LT is incremented as BLK is being filled with compressed data to be output to display file.

NBJ Number of cells in current band b_j.

NBM Number of cells on band north of current band b_{j-1}.

NBP Number of cells on band south of current band b_{j+1}.

NB1 Holds b_j for first dummy band, NB1 = 1.

NB2 Holds b_j for first data band, J = 2.

NB3 Holds b_j for second data band, J = 3.

NUN Holds coded ν_N for current cell column.

NUS Holds coded ν_S for current cell column.

RAP P_* tendency, $\frac{\partial P_*}{\partial t}$ for current cell column.

RT Real running time. If RT $\geq$ RTL - 10, end processing is initiated.

RTE Tendencies of 3-D prognostic variables for each of the KV levels indexed (I,J).

WEWJ East-west weight for current band $W_E = W_W$ (KMI).

WNJ North weight $W_{\Delta\lambda}N_{j-1} = \frac{\Delta l}{\Delta A_j}$ for current band (KMI), where Δl is the length of the north coincident boundary segment, and, $\Delta A_j = 2a^2 \Delta\lambda_j \cos\theta_j \sin\left(\Delta\theta_j/2\right)$ is constant for each band.

WNSJ $W_N - W_S$, where $W_N = \sum_N W_l$, and $W_S = \sum_S W_l$. Value appears in G_θ operator (KH).

WSJ South weight $W_{\Delta\lambda}{}^{'}S_{j+1} = \Delta l/\Delta A_j$, for current band.

WXN Holds W_{η_1}, W_{η_2} for current cell column, WXN(1) = W_{η_1}, WXN(2) = W_{η_2}.

WXS Holds $W_{\xi 1}$, $W_{\xi 2}$ for current cell column, WXS(1) = $W_{\xi 1}$, WXS(2) = $W_{\xi 2}$.

<u>Helpful Comments</u>

<u>LT = 0 $ IF (ISS .EQ. 0) GO TO 1011 $ WRITE (IPRT, 1012) NTS $ GO TO 1007</u>

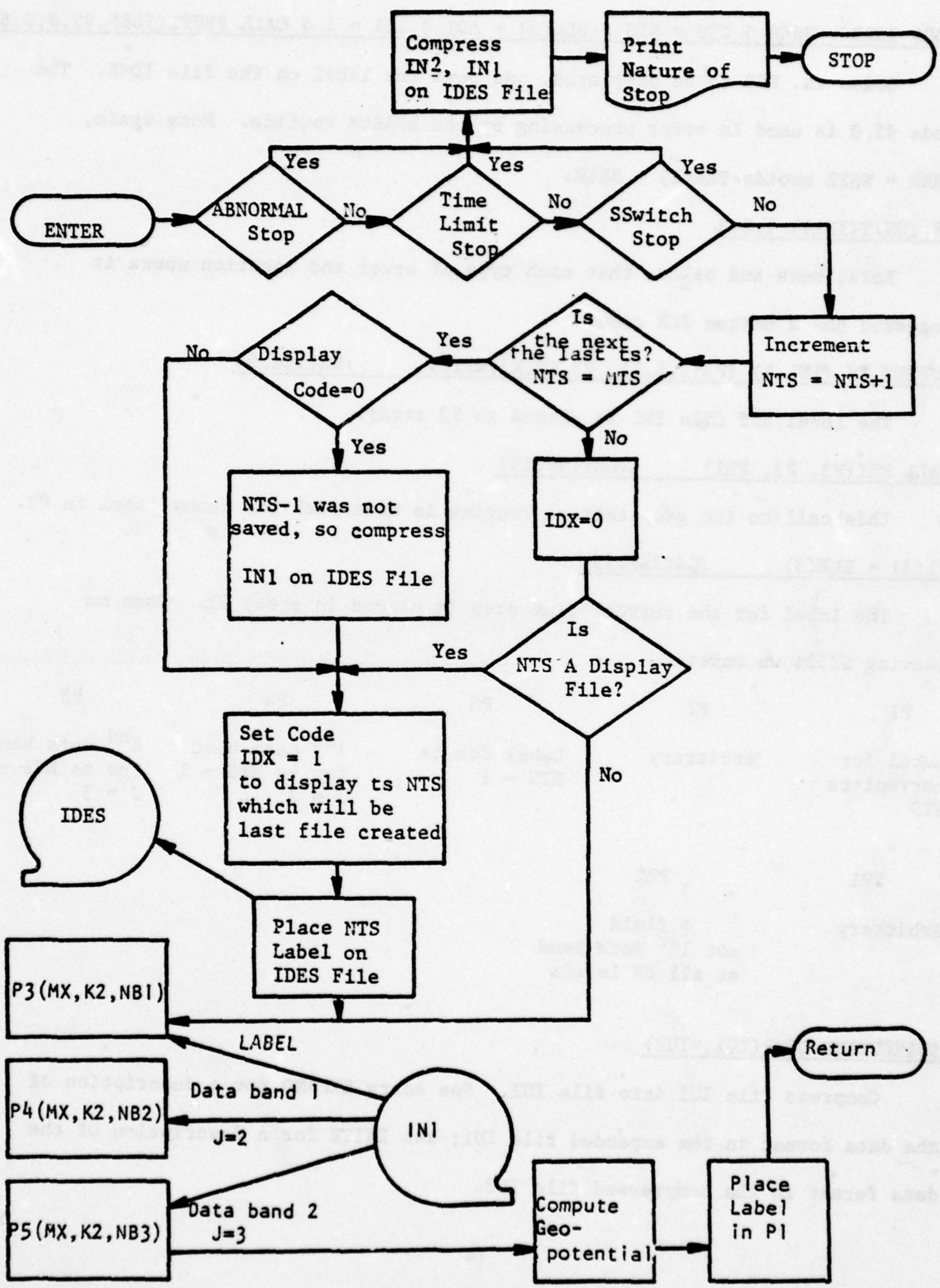
On first entering STISϕ, ISS has been preset zero in GEX. On subsequent entries ISS may have been set to 1, if TISϕ has detected a negative value in the P_* scalar field.

<u>1007 U = UNIT(IN1) + UNIT(IN2) $ REWIND IN1 $ REWIND IN2 $ CALL COMP(IN2,IDES)</u>

If any one of the stop conditions exist then IN1 and IN2 are compressed on the display file. Since the file names were rotated in GEX before the entry to STISϕ, IN1 contains the last time step created, and IN2 the previous one. Note that the display file counter is incremented two. The subroutine COMP also has the entry expand.

1001 NTS = NTS + 1 $ ADT = ADT + DT $ IF (NTS .NE. MTS) GO TO 1006 $
IF (IDX .NE. 0 GO TO 1009

None of the conditions for an immediate stop has been met. Since the next time step will be created, NTS is incremented. A test is made, to see if this will be the last time step. If so, both the ts. NTS and NTS - 1 must be saved on display, though neither may be a regular display file. This is necessary, since we need two consecutive steps for restart. Thus, a test is made to see if NTS was already saved (IDX = 1). If not, IN1 is compressed on display <u>and</u> IOT is coded to be displayed (IDX = 1). Otherwise, IOT is coded to be displayed, and its label sent out on IDES file.



ENTER
ABNORMAL Stop
No
Yes
Time Limit Stop
Yes
SSwitch Stop
Yes
No
Compress IN2, IN1 on IDES File
Print Nature of Stop
STOP
Increment NTS = NTS+1
Is the next the last ts? NTS = MTS
Yes
No
Display Code=0
No
Yes
NTS-1 was not saved, so compress IN1 on IDES File
IDX=0
Is NTS A Display File?
Yes
No
Set Code IDX = 1 to display ts NTS which will be last file created
IDES
Place NTS Label on IDES File
P3(MX,K2,NB1)
LABEL
P4(MX,K2,NB2)
Data band 1
J=2
IN1
P5(MX,K2,NB3)
Data band 2
J=3
Compute Geo-potential
Place Label in P1
Return

1009 ISER = NSER $ ITS = NTS $ BLK(3) = ADT $ IDX = 1 $ CALL PBUFT(IDES,95,0,U,BLK3)

Codes ts. NTS to be displayed, and puts the label on the file IDES. The code 95,0 is used in error processing by the BERROR routine. Note again, ISER = NSER avoids BLK(1) = NSER.

IF (UNIT(IN1)) 1,2,3

Note, here and below, that each type of error and location where it occurred has a unique ICE code.

BUFFER IN (IN1,1) (P3(1,1,1), P3(NX,K2,NB2)) (GSTISE.9)

The label off file IN1 is placed in P3 array.

CALL PH1(P3, P3, PH1) (GSTISE.19)

This call to the geopotential routine is made with the dummy label in P3.

P1(3) = BLK(3) (GSTISE.25)

The label for the current time step is placed in array P1. Then on leaving STISϕ we have:

P1	P2	P3	P4	P5
Label for current ts NTS	Arbitrary	Label for ts NTS - 1	1^{st} data band for ts NTS - 1 J = 2	2^{nd} data band for ts NTS - 1 J = 3

PH1	PH2
Arbitrary	ϕ field for 1^{st} data band at all KV levels

SUBROUTINE COMP(IU1, IU2)

Compress file IU1 into file IU2. See entry EXPAND for a description of the data format in the expanded file IU1; see INITE for a description of the data format in the compressed file IU2.

Common Blocks

/UNITS/

/AC/

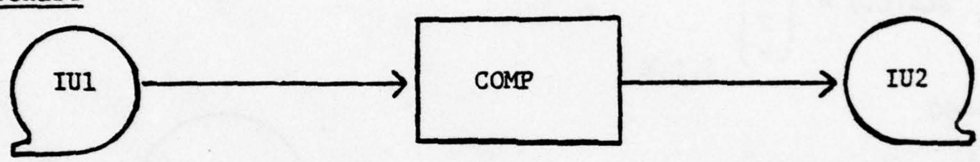
/BUF/

/ /

Variable Description

See ENTRY EXPAND.

Flowchart



Helpful Comments

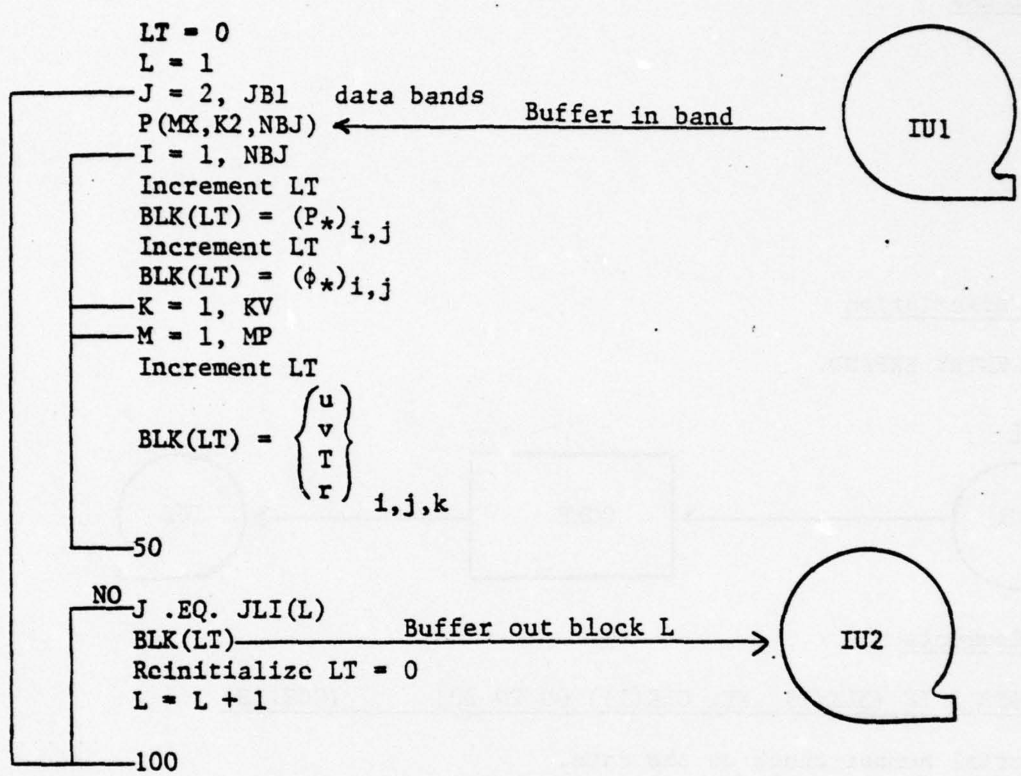
ISER = NSER $ IF (BLK(1) .NE. CLK(1)) GO TO 201 (GCE.12)

A serial number check on the data.

BUFFER OUT (IU2,1) BLK(1), BLK(3)) (GCE.15)

Label goes out on IU2, exactly as it came in on IN1.

The loop begins with the expanded file IU1 positioned after the label, and before the first data band. The logic of the loop is:



LT = 0 $ L = 1 $ DO J= 2, JB1 $ NBJ = NB(J) (GCE.18)

LT is reset to zero whenever BLK is to be refilled with compressed data. L keeps count of blocks being created, the total number of which was predetermined in the INIT run.

IF (J .LT. JLI(L)) GO TO 100 (GCE.26)

Go to 100 if this band is not the last latitude in a compressed block; otherwise, output the data, since the next band in will exceed the I/O size.

SUBROUTINE CBUFN (IU, I, J, U, A, N)

ENTRY PBUFN

ENTRY CBUFT

ENTRY PBUFT

N words of data are buffered in (subroutine name ends in N), from IU; or, buffered out (subroutine name ends in T) to IU. If prefix is C, unit

checks are done on entry and on exit, if prefix is P, unit checks are done only on entry. CODE is passed to BERROR if buffer error occurs.

SUBROUTINE PHI(PJ, PJS, PHJ)

Returns geopotential scalar field for band j at all KV levels. The value of geopotential is given at the top and bottom of a box. Subroutine is now programmed for 1 level. In the general case PHJ should be dimensioned PHJ(KV1,NBJX).

Common Blocks

/BUF/

/UNITS/

/AC/

/CEH/

Helpful Comments

```
PHJ(KV1,I) = GO * PJ(1,K1,I)
1  PHJ(1,I) = PHJ(KV1,I)          (GPHIBR.10)
```

$$\text{From } \phi_{k\pm\frac{1}{2}} = \phi_k \mp \frac{\Delta_k\sigma}{2} \cdot \frac{RT_k}{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})} \quad \text{(KH)}$$

we have

$$\phi_{k-\frac{1}{2}} = \phi_{k+\frac{1}{2}} + \Delta\sigma_k \frac{RT_k}{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})}$$

Since we carry ϕ_* field ($\phi_* = \phi_{kv+\frac{1}{2}}$), we can determine the ϕ field iteratively through the top of the atmosphere ($\phi_{\frac{1}{2}}$). In the special case here programmed, $T_k \equiv 0$ in space and time, so that

$$\phi_{k-\frac{1}{2}} = \phi_{k+\frac{1}{2}}$$

PHJ(K,I) = $\phi_{k-\frac{1}{2}}$ $\Longrightarrow$ PHJ(KV1,I) = $\phi_{kv+\frac{1}{2}} = g_oH$

where H is the height of the free surface, and is carried in PJ(1,K1,I).

Also, PHJ(1,I) = $\phi_{\frac{1}{2}} = g_oH$

Note: calls to PHI(PJ,PJS,PHJ) with dummy bands as PJ, return data of no significance in PHJ.

SUBROUTINE ROTATE (TIS)

Rotation Subprogram. This routine rotates the data arrays from the north pole to the south pole in the process of updating the prognostic variables to the next time step, with a chosen procedure subroutine TIS. The logical rotation of the data avoids the actual transfer of data in main memory. When the data units are left rotated, they assume the proper configuration for updating the next latitude band j + 1. A maximum of 12 prognostic arrays and 8 diagnostic arrays is necessary to accomodate the most complex of the TIS's.

Common Blocks

/ / Holds arrays which are being rotated.

/BUF/

/UNITS/

/AC/

/BC/ Holds index of current latitude band.

Variable Description

No new variables introduced.

Helpful Comments

```
    J = 1
36  J = J + 1        (GROTATE.28)
```

The first band processed will be index J = 2 (the first data band). A return to 36 will be made if a rotation back to the original order is called for, and not all the data bands have been processed. Since the least common multiple [12, 8] = 24, this will occur after 23 rotations (at statement GROTATE.147).

CALL TIS(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12, PH1,PH2,PH3,PH4,PH5,PH6,
1 PH7,PH8) (GROTATE.30)

On the first pass, performs time integration on the first data band (J = 2). The P and PH arrays contain that information indicated at the end of STIS.

IF (J - JB1) 1, 2, 2

Tests if all data bands have been processed. If not, latitude index J is incremented, another rotation of data arrays is performed, and the next band is processed. If all bands have been processed, the subroutine ENTIS is called to end processing on this time step, by putting out the last updated data band, which should reside in array P2, and also putting out the dummy bands. Then a return to GEX is called.

SUBROUTINE TISϕ

Euler time integration scheme. This routine updates the information of latitude band J to time N.

Common Blocks

/LTC/

/ /

/BUF/

/UNITS/

/AC/

/FG/ Holds parameters for line integration around cell, for all KV cells of current (i,j).

/BC/

/CEF/

/CEFG/

Variable Description

DA1 = P4(1,K2,I) Contains both η_2 and ν_N code for current index (i,j).

$$[DA1]^{INT} = \eta_{2,i,j} \text{ and } \nu_N = DA1 - [DA1]^{INT}$$

DA2 = P4(2,K2,I) Contains both ξ_2 and ν_S code for current index (i,j).

$$[DA2]^{INT} = \xi_{2,i,j} \text{ and } \nu_S = DA2 - [DA2]^{INT}$$

IUN Holds file name, on which buffer error occurred.

K Level index.

L Zonal index of cells contiguous to current cell (i,j) as line integration proceeds along cell boundary in RHS.

M Index of 3-D prognostic variable in P array.

MB Code passed by RHS to line integration routines.
- MB = 1 Integrate on east-west boundary
- MB = 2 Integrate on north-south boundary

N Do loop index for putting out display file.

NBJM1 b_{j-1}.

NBJP2 b_{j+2}.

PH Array holds $\frac{P_{*o} + P_{*l}}{2} (\phi_{l_{k-\frac{1}{2}}} - \phi_{o_{k-\frac{1}{2}}})$ for all KV1 level boundaries, and current index (i,j). Returned from subroutine PSP, which is called by RHS.

PST N - 1 value of P_* for current (i,j).

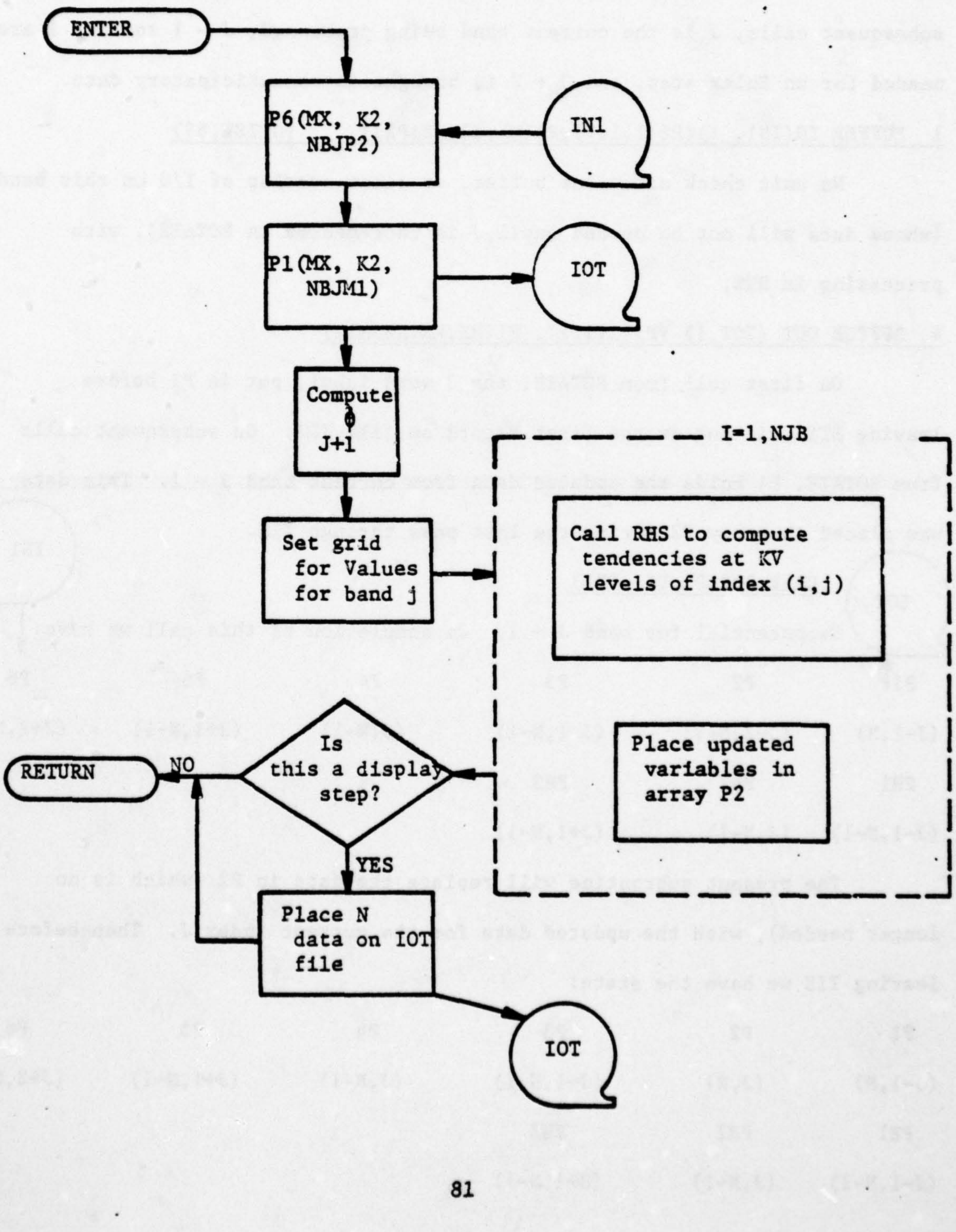
PST1 Updated (N) value of P_* for current index (i,j).

PT Array holds $\frac{P_{*o} + P_{*l}}{2} \frac{T_o + T_l}{2}$ for all KV levels and current index (i,j). PT is returned from subroutine PST, which is called by RHS.

PY Array holds $\frac{P_{*o} + P_{*l}}{2} \frac{\psi_o + \psi_l}{2}$ for all KV levels and current index (i,j). Here ψ is one of the 3-D prognostic variables. PY is returned from subroutine PSY, which is called by RHS.

YP Array holds $\frac{P_{*o} + P_{*l}}{2} \frac{Y_o + Y_l}{2} \frac{\psi_o + \psi_l}{2}$ for all KV levels, for current index (i,j), and each possible 3-D variable ψ. Y is u along east-west boundary, and v along the north-south boundary. YP is returned from subroutine YSL, which is called by RHS.

Y1 Array holds $\frac{P_{*o} + P_{*l}}{2} \frac{Y_o + Y_l}{2}$ for all KV levels for current index (i,j). Y is u along east-west boundary, and v along north-south boundary. Y1 is returned from subroutine YSL, which is called by RHS.



Helpful Comments

NBJP2 = NB(J+2)
NBJM1 = NB(J-1)   (GTISE.66)

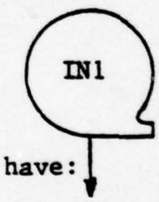
On first call from ROTATE, J = 2 (first data band). The J - 1 is the index of the dummy band, and J + 1 is the index of the anticipatory band. On subsequent calls, J is the current band being processed, J - 1 and J + 1 are needed for an Euler step, and J + 2 is brought in as anticipatory data.

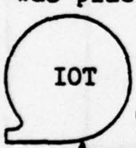
1 BUFFER IN(IN1, 1)(P6(1,1,1),P6(MX,K2,NBJP2))   (GTISE.69)

No unit check after the buffer, to allow overlap of I/O on this band (whose data will not be needed until J is incremented in ROTATE), with processing in RHS.

4 BUFFER OUT (IOT,1) (P1(1,1,1), P1(MX,K2,NBJM1))

On first call from ROTATE, the 3 word label, put in P1 before leaving STIS, is put as the first record on file IOT. On subsequent calls from ROTATE, P1 holds the updated data from current band J - 1. This data was placed in array P2 during the last pass through TIS.





CALL PHI(P5,P5, PH3)

Geopotential for band J + 1. On completion of this call we have:

P1	P2	P3	P4	P5	P6
(J-1,N)	(J-2,N-1)	(J-1,N-1)	(J,N-1)	(J+1,N-1)	(J+2,N-1)
PH1	Ph2	PH3			
(J-1,N-1)	(J,N-1)	(J+1,N-1)			

The present subroutine will replace the data in P2 (which is no longer needed), with the updated data for the current index J. Then before leaving TIS we have the state:

P1	P2	P3	P4	P5	P6
(J-1,N)	(J,N)	(J-1,N-1)	(J,N-1)	(J+1,N-1)	(J+2,N-1)
PH1	PH2	PH3			
(J-1,N-1)	(J,N-1)	(J+1,N-1)			

ROTATE then increments J, and rotates the array names. Then on return to TIS we have J + 1 as the current index, and after the call to PHI

	P1	P2	P3	P4	P5	P6	
IOT ←	(J,N)	(J-1,N-1)	(J,N-1)	(J+1,N-1)	(J+2,N-1)	(J+3,N-1)	← IN1
	PH1	PH2	PH3				
	(J,N-1)	(J+1,N-1)	(J+2,N-1)				

This is exactly the beginning state for the current index J + 1.

WNSJ = DWNS(J)   (GTISE. 87)

COP, COT, WEWJ, WSJ, WNJ, WNSJ are grid parameters for the current index J, and are independent of i, k. They are set outside the loop.

ITA(2) = P4(1,K2,IM)

IXI(2) = P4(2,K2,IM)

WXN(2) = P3(3,K1,IM)

WXS(2) = P4(3,K2,IM)   (GTISE.95)

Initialize $\eta_{2,b_j,j}$, ξ_{2,b_j,b_j} $W_{\Delta\lambda}{}^{N}_{2,b_j,j}$, and $W_{\Delta\lambda}{}^{S}_{2,b_j,j}$ for i = 1 ($i - 1 \equiv b_j$).

IF (I - NBJ) 7, 8, 9

7 Current cell has cells to east yet to be processed

8 Current cell is last cell on band to be processed. East contiguos cell has index i = 1.

9 All cells on band have been processed.

(i) ITA(1) = ITA(2)
(ii) IXI(1) = IXI(2)
(iii) WXN(1) = WNJ - WXN(2)
(iv) WXS(1) = WSJ - WXS(2)
(v) WXN(2) = P4(3,K1,I)
(vi) WXS(2) = P4(3,K2,I)
(vii) DA1 = P4(1,K2,I)
(viii) DA2 = P4(2,K2,I)   (GTISE.107)

That is (i) $\eta_{1,i,j} = \eta_{2,i-1,j}$

(ii) $\xi_{1,i,j} = \xi_{2,i-1,j}$

(iii) $W_{\Delta\lambda}{}^{N}_{1,i,j} = W_{\Delta\lambda}{}^{N}_{2,i-1,j}$ (KMI)

(iv) $W_{\Delta\lambda}S_{1,i,j} = W_{\Delta\lambda}S_{j+1} - W_{\Delta\lambda}S_{2,i-1,j}$

By this method we avoided carrying <u>both</u> the east and west values of η, ξ, W_η, W_ξ in the P arrays

(v) $W_{\Delta\lambda}N_{2,i,j}$

(vi) $W_{\Delta\lambda}S_{2,i,j}$ } For current index i.

(vii) DA1 The packed word η, ν_N.

(viii) DA2 The packed word ξ, ν_S.

Then $\eta_{2,i,j} = [DA1]^{INT}$

$\xi_{2,i,j} = [DA2]^{INT}$ } For current index i.

and

$[DA1]^{MOD(1.)}$ = decimal part of word

$[DA2]^{MOD(1.)}$ = decimal part of word

$\nu_N = 10.\ [DA1]^{MOD(1.)} + .5$

$\nu_S = 10.\ [DA2]^{MOD(1.)} + .5$ } For current index i.

<u>CALL RHS (P4,P3,P5,PH2,PH1,PH3) (GTISE.112)</u>

Computes the tendencies for current index (i,j) for all KV levels. These tendencies are returned in CEF common in RAP and RTE.

```
  (i)  PST = P4(1,K1,I)
 (ii)  PST1 = PST + DT * RAP
(iii)  IF (PST .LE. 0) ISS = 1
 (iv)  P2(1,K1,I) = PST1                 (GTISE.117)
```

That is

(i) P_* for current index (i,j) at ts. N - 1.

(ii) Updated $(P_*)_N = (P_*)_{N-1} + \Delta t \frac{\partial P_*}{\partial t}$

(iii) Sets error code if P_* for any column becomes negative.

(iv) Places updated P_* in array P2 for subsequent output to IOT file.

<u>P2(M,K,I) = (PST * P4(M,K,I) + DT * RTE(M,K))/PST1</u>

$$\psi_N = \left\{\frac{(P_*)_{N-1}\ \psi_{N-1} + \Delta t \frac{\partial\psi}{\partial t}}{(P_*)_N}\right\}$$

where ψ is one of the 3-D prognostic variables. Note replacement of data in P2 which is no longer needed.

P2(2,K2,I) = DA2 (GTISE.127)

All grid values remain the same for ts. N as for ts. N - 1.

IM = I
I = I + 1 (GTISF.129)

Process next eastward cell.

IF (IDX. EQ. 0) GO TO 1010 $ DO 1001 N = 1, NBJ

If this is not a display step, return to ROTATE. If it is a display step, place updated, band j data, in compressed format into the array BLK. Then check to see if current band index J, is the last band of a compressed data block. If it is not, return to ROTATE. If it is, then buffer BLK onto display file, reinitialize the LT index for BLK, and increment the counter for the number of I/O blocks processed (LB). Then return to ROTATE.

15 CALL BERROR(J,ICE,IUN)

ICE identifies the routine, J gives the current index when the error occurred.

Note: We only present the EULER routines STISϕ, TISϕ. Other integration routines will differ in detail, but not in substance from that presented here.

SUBROUTINE RHS(PJ,PJM,PJP,PHJ,PHM,PHP)

This subroutine returns for column (i,j) the right hand side of each of the prognostic equations. The tendencies are passed through CEF common.

Common Blocks

/BUF/

/UNITS/

/AC/

/FG/

/CEF/

/CEH/

/CEFG/

/FM/ Holds forcing terms for KV levels indexed (i,j).

Variable Description

CORA Array returns condensation rate for all KV levels from CON entry of forcing function subprogram. Presently, values returned are zero.

CPRT P_{*o}/c_p

ELPP Array of $L_\lambda(P_*\phi)$ at the KV1 level boundaries (KH).

ETPP Array of $L_\theta(P_*\phi)$ at the KV1 level boundaries.

FID Array returns $(F_\lambda, F_\theta, F_T, F_t)$ for all KV levels from forcing function subprogram (KH). Presently, values returned are zero.

GLP $G_\lambda(P_*)$

GLPT Array of $G_\lambda(P_*T)$ at all KV levels of current index (i,j).

GTP $G_\theta(P_*)$

GTPT $G_\theta(P_*T)$

H Array holds $H_1(Y)$ or $H(K) = \sum_{k=1}^{k} H_1(l)\Delta\sigma_k$ (KH)

IT Index for η, ξ arrays.
IT = 1 west side
IT = 2 east side

KVV KV. Avoids control of do loop in common.

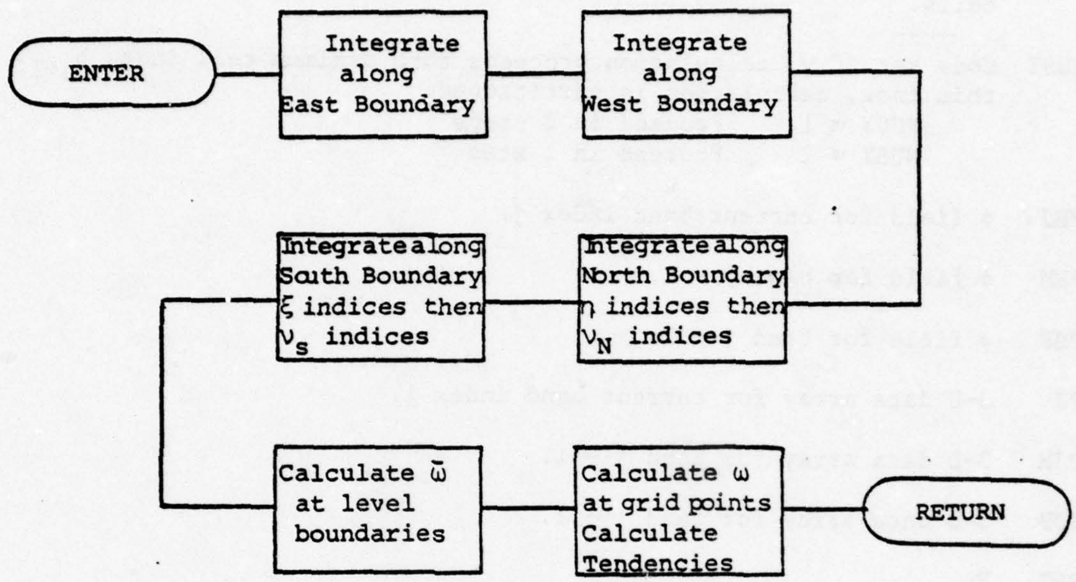
M Index of 3-D prognostic variable.

NU Index of do loop for north ν sum, and south ν sum.

NUL Lower zonal index for calculation of line integral along ν contiguous cells.

NUNY Code set if ν_N calculation proceeds thru maximum cell index b_{j-1}. In this case, calculation is partitioned.
NUNY = 1 Proceed in 2 steps
NUNY = 2 Proceed in 1 step

NUP	Upper zonal index for calculation of line integral along ν contiguous cells.
NUSY	Code set if ν_S calculation proceeds thru maximum cell index b_{j+1}. In this case, calculation is partitioned. NUSY = 1 Proceed in 2 steps NUSY = 2 Proceed in 1 step
PHJ	ϕ field for current band index j.
PHM	ϕ field for band j - 1.
PHP	ϕ field for band j + 1.
PJ	3-D data array for current band index j.
PJM	3-D data array for band j - 1.
PJP	3-D data array for band j + 1.
PRT	P_{*o}
QDT	Array returns added heat for all KV levels from HEAT entry of forcing function subprogram. Presently, values returned are zero.
T1	$P_{*o}(\hat{f} + \hat{m}u_k)$
T2	$\sigma_{k+\frac{1}{2}} L_\lambda(P_*\phi_{k+\frac{1}{2}}) - \sigma_{k-\frac{1}{2}} L_\lambda(P_*\phi_{k-\frac{1}{2}})$
T3	$\sigma_{k+\frac{1}{2}} L_\theta(P_*\phi_{k+\frac{1}{2}}) - \sigma_{k-\frac{1}{2}} L_\theta(P_*\phi_{k-\frac{1}{2}})$
WT	Temporary variable to hold $W_{\Delta\lambda}{}^{N}_{1,i,j}$ $(W_{\Delta\lambda}{}^{S}_{1,i,j})$ or $W_{\Delta\lambda}{}^{N}_{2,i,j}$ $(W_{\Delta\lambda}{}^{S}_{2,i,j})$.
Z	Array holding vertical P velocity ω at all 25 levels of current column. ω is only needed diagnostically, and is not held outside RHS.
ZB1	$\dfrac{1}{2(\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}})} P_{*o}\,\overline{\omega}_{k-\frac{1}{2}}$ (KH)
ZB2	$\dfrac{1}{2(\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}})} P_{*o}\,\overline{\omega}_{k+\frac{1}{2}}$ (KH)



Helpful Comments

MB = 1 (GRHS.43)

Line integral around cell begins with east-west boundary. These are the simplest to process since $W_W = W_E$.

GLP = .5 * WEW * (PH(1,K1,IP) - PJ(L,K1,JM)) $ L = IP

$$G_\lambda(P_*) = W_{EW}\left\{\frac{P_{*E} - P_{*W}}{2}\right\}$$

where P_{*E} is the value of P_* in the east contiguous cell, (I = IP) and P_{*W} is the value of P_* in the west contiguous cell (I = IM). The contiguous cell index (L), is then set to IP in anticipation of the calls to YSL, PST, and PSP. L is passed to these routines in FG common.

CALL YSL(PJ,PT)

Returns Y1 array, holding $\left(\frac{P_{*o} + P_{*E}}{2}\right)\left(\frac{u_o + u_E}{2}\right)$ for all KV levels, and YP array holding $\left(\frac{P_{*o} + P_{*E}}{2}\right)\left(\frac{u_o + u_E}{2}\right)\left(\frac{\psi_o + \psi_E}{2}\right)$ for all KV levels. The formal

arguments PJ, PJ are the same since the contiguous cell is on the same band as the current cell.

CALL PST(PJ,PJ)

Return PT array holding $\left(\frac{P_{*O} + P_{*E}}{2}\right)\left(\frac{T_O + T_E}{2}\right)$ for all KV levels.

CALL PSP(PJ,PJ,PHJ,PHJ)

Returns PH array holding $\left(\frac{P_{*O} + P_{*E}}{2}\right)\left(\phi_{E_{k-\frac{1}{2}}} - \phi_{O_{k-\frac{1}{2}}}\right)$ for all KV1 level boundaries.

1 CONTINUE (GRHS.59)

On emerging from the k indexed loop

$$G_\lambda(P_*T) = \frac{P_{*O} + P_{*E}}{2} \frac{T_O + T_E}{2}$$

$$L_\lambda(P_*\phi) = \frac{P_{*O} + P_{*E}}{2} (\phi_{E_{k-\frac{1}{2}}} + \phi_{O_{k-\frac{1}{2}}})$$

$$H(K) = \frac{P_{*O} + P_{*E}}{2} \left(\frac{u_O + u_E}{2}\right)$$

$$D3(M,K) = \frac{P_{*O} + P_{*E}}{2} \frac{u_O + u_E}{2} \frac{\psi_O + \psi_E}{2}$$

ELPP(KV1) = PH(KV1) (GRHS.61)

$$L_\lambda(P_*\phi)_{kv1} = \frac{P_{*O} + P_{*E}}{2} (\phi_{E_{kv+\frac{1}{2}}} - \phi_{O_{kv+\frac{1}{2}}})$$

L = IM (GRHS.63)

Integration proceeds along the western boundary of a box. Now YSL returns the array Y1 containing $\frac{P_{*O} + P_{*W}}{2} \frac{u_O + u_W}{2}$ at all KV levels, and the array YP containing $\frac{P_{*O} + P_{*W}}{2} \frac{u_O + u_W}{2} \frac{\psi_O + \psi_W}{2}$ at all KV levels, and PSP returns the PH array holding $\left(\frac{P_{*O} + P_{*W}}{2}\right)(\phi_{W_{k-\frac{1}{2}}} - \phi_{O_{k-\frac{1}{2}}})$ for all KV1 level boundaries.

3 CONTINUE (GRHS.78)

On emerging from the k indexed loop

$$G_\lambda(P_*T)_k = W_{\underset{J}{EW}} \left\{ \frac{P_{*O} + P_{*E}}{2} \frac{T_O + T_E}{2} - \frac{P_{*O} + P_{*W}}{2} \frac{T_O + T_W}{2} \right\}$$

$$L_\lambda(P_*\phi)_k = W_J^{EW}\left\{\frac{P_{*o}+P_{*E}}{2}(\phi_{E_{k-\frac12}}-\phi_{o_{k-\frac12}}) - \frac{P_{*o}+P_{*W}}{2}(\phi_{W_{k-\frac12}}-\phi_{o_{k-\frac12}})\right\}$$

Note: the factor ½ is absent from L_λ (2·9 KH).

$$H(K) = W_J^{EW}\left\{\frac{P_{*o}+P_{*E}}{2}\,\frac{u_o+u_E}{2} - \frac{P_{*o}+P_{*W}}{2}\,\frac{u_o+u_W}{2}\right\}.$$

$$D3(M,K) = W_J^{EW}\left\{\frac{P_{*o}+P_{*E}}{2}\left(\frac{u_o+u_E}{2}\right)\frac{\psi_o+\psi_E}{2} - \frac{P_{*o}+P_{*W}}{2}\left(\frac{u_o+u_W}{2}\right)\frac{\psi_o+\psi_W}{2}\right\}$$

ELPP(KV1) = WEWJ * {ELPP(KV1) - PH(KV1)} (GRHS.80)

$$L_\lambda(P_*\phi)_{kv1} = W_J^{EW}\left\{\frac{P_{*o}+P_{*E}}{2}(\phi_{E_{kv+\frac12}}-\phi_{o_{kv+\frac12}}) - \frac{P_{*o}+P_{*W}}{2}(\phi_{W_{kv+\frac12}}-\phi_{o_{kv+\frac12}})\right\}$$

MB = 2 (GRHS.88)

Integration proceeds along the north boundary. The easternmost ($L = \eta_1$) contiguous cells are handled first. Notice in the calls to YSL, PST, and PSP, that PJM identifies the northern contiguous band on which the adjacent cells indexed L are located. For example YSL returns $\frac{P_{*o}+P_{*\eta_1}}{2}\;\frac{v_o+v_{\eta_1}}{2}$ in Y1, and $\frac{P_{*o}+P_{*\eta_1}}{2}\;\frac{v_o+v_{\eta_1}}{2}\;\frac{\psi_o+\psi_{\eta_1}}{2}$ in YP on the first pass through the loop, and, $\frac{P_{*o}+P_{*\eta_2}}{2}\;\frac{v_o+v_{\eta_2}}{2}$ in Y1, and $\frac{P_{*o}+P_{*\eta_2}}{2}\;\frac{v_o+v_{\eta_2}}{2}\;\frac{\psi_o+\psi_{\eta_2}}{2}$ in YP on the second pass.

5 CONTINUE (GRHS.114)

On emerging from the loop

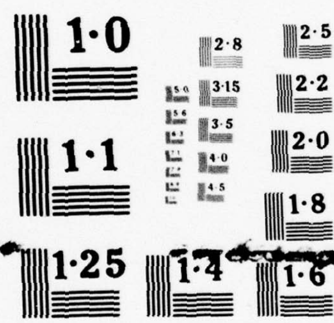
$$G_\lambda(P_*) = W_{\eta_1}P_{*\eta_1} + W_{\eta_2}P_{*\eta_2}$$

$$G_\theta(P_*T)_k = \frac{P_{*o}+P_{*\eta_1}}{2}\,\frac{T_o+T_{\eta_1}}{2}\,W_{\eta_1} + \frac{P_{*o}+P_{*\eta_2}}{2}\,\frac{T_o+T_{\eta_2}}{2}\,W_{\eta_2}$$

AD-A048 481 REGIS COLL WESTON MASS F/G 4/2
A GENERALIZED COMPUTER PROGRAM FOR PRIMITIVE-EQUATION MODELS.(U)
MAY 77 T J LEONARD, J C METTAUER F19628-77-C-0010
UNCLASSIFIED SCIENTIFIC-1 AFGL-TR-77-0183 NL

2 OF 2
AD
A048481

END
DATE
FILMED
2-78
DDC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

$$L_\theta(P_*\phi)_k = \frac{P_{*o} + P_{*\eta_1}}{2} (\phi_{\eta_1} - \phi_o)_{k-\frac{1}{2}} W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2} (\phi_{\eta_2} - \phi_o)_{k-\frac{1}{2}} W_{\eta_2}$$

$$H(K) = W_J^{EW} \left\{ \frac{P_{*o} + P_{*E}}{2} \frac{u_o + u_E}{2} - \frac{P_{*o} + P_{*W}}{2} \frac{u_o + u_W}{2} \right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2} \frac{v_o + v_{\eta_1}}{2} W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2} \frac{v_o + v_{\eta_2}}{2} W_{\eta_2}$$

$$D3(M,K) = W_J^{EW} \left\{ \frac{P_{*o} + P_{*E}}{2} \frac{u_o + u_E}{2} \frac{\psi_o + \psi_E}{2} - \frac{P_{*o} + P_{*W}}{2} \frac{u_o + u_W}{2} \frac{\psi_o + \psi_W}{2} \right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2} \frac{v_o + v_{\eta_1}}{2} \frac{\psi_o + \psi_{\eta_1}}{2} W_{\eta_1}$$

$$+ \frac{P_{*o} + P_{*\eta_2}}{2} \frac{v_o + v_{\eta_2}}{2} \frac{\psi_o + \psi_{\eta_2}}{2} W_{\eta_2}$$

GO TO (8,9,10,11,12,13) NUN

The coded value of ν_N is used to determine the nature of the integration between the extreme contiguous cells. Branch to:

8 Corresponds to case $\nu = 0$ (KMI) or $\nu_N = 1$. In this case the westernmost and easternmost cells are adjacent, so that the northern integration is already complete.

9 Corresponds to case $\nu = -1$ or $\nu_N = 2$. The westernmost and easternmost cells coincide. Its necessary then to subtract the value above where this cell is counted twice, and, then, northern integration is complete. Notice in this case that $W_{\eta_1} = W_{\eta_2} = W_{\Delta\lambda}{}_J^N$.

10 In this case, there are $\nu = \eta_2 - (\eta_1+1)$ cells between the cell indexed η_1 and the cell indexed η_2. These cells are indexed $\eta_1 + 1$ thru $\eta_2 - 1$.

11 In this case, $\eta_1 = b_{j-1}$ and $\eta_2 < b_{j-1}$. There are $\nu = \eta_2 - 1$ cells between the cell indexed η_1 and the cell indexed η_2. These cells are indexed 1 thru $\eta_2 - 1$.

12 In this case, $\eta_2 = 1$ and $\eta_1 < b_{j-1}$. There are $\nu = b_{j-1} - \eta_1$ cells between the cell indexed η_1 and the cell indexed η_2. These cells are indexed $\eta_1 + 1$ thru b_{j-1}.

13 In this case $\eta_1 < b_{j-1}$ and $\eta_2 > 1$. This integration is handled is two stages. First (NUNY=1) the $\nu = b_{j-1} - \eta_1$ cells between the cell indexed η_1 and the cell indexed 1 are processed. These cells are indexed $\eta_1 + 1$ thru b_{j-1}. Then (NUNY=2) the $\nu = \eta_2 - 1$ cells between the cell indexed b_{j-1} and the cell indexed η_2 are processed. These cells are indexed 1 thru $\eta_2 - 1$.

8 CONTINUE (GRHS.171)

On emerging from the north NU sum we have

$$G_\theta(P_*) = W_{\eta_1}P_{*\eta_1} + W_{\eta_2}P_{*\eta_2} + \sum_{\nu_N} W_l^N P_{*l}$$

$$G_\theta(P_*T)_k = \frac{P_{*o} + P_{*\eta_1}}{2}\,\frac{T_o + T_{\eta_1}}{2}\,W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2}\,\frac{T_o + T_{\eta_2}}{2}\,W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2}\,\frac{T_o + T_l}{2}$$

$$L_\theta(P_*\phi)_k = \frac{P_{*o} + P_{*\eta_1}}{2}(\phi_{\eta_1} - \phi_o)_{k-\frac{1}{2}}\,W_{\eta_1}$$

$$+ \frac{P_{*o} + P_{*\eta_2}}{2}(\phi_{\eta_2} - \phi_o)_{k-\frac{1}{2}}\,W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2}(\phi_l - \phi_o)_{k-\frac{1}{2}}$$

$$H(K) = W_J^{EW}\left\{\frac{P_{*o} + P_{*E}}{2}\,\frac{u_o + u_E}{2} - \frac{P_{*o} + P_{*W}}{2}\,\frac{u_o + u_W}{2}\right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2}\,\frac{v_o + v_{\eta_1}}{2}\,W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2}\,\frac{v_o + v_{\eta_2}}{2}\,W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2}\,\frac{v_o + v_l}{2}$$

$$D3(M,K) = W_J^{EW}\left\{\frac{P_{*o} + P_{*E}}{2}\,\frac{u_o + u_E}{2}\,\frac{\psi_o + \psi_E}{2} - \frac{P_{*o} + P_{*W}}{2}\,\frac{u_o + u_W}{2}\,\frac{\psi_o + \psi_W}{2}\right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2}\,\frac{v_o + v_{\eta_1}}{2}\,\frac{\psi_o + \psi_{\eta_1}}{2}\,W_{\eta_1}$$

$$+ \frac{P_{*o} + P_{*\eta_2}}{2}\,\frac{v_o + v_{\eta_2}}{2}\,\frac{\psi_o + \psi_{\eta_2}}{2}\,W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2}\,\frac{v_o + v_l}{2}\,\frac{\psi_o + \psi_l}{2}$$

C SOUTH BOUNDARY (GRHS.173)

The south boundary integration proceeds exactly as the north. Note in particular, that the contiguous band in this case is PJP.

24 CONTINUE (GRHS.256)

On emerging from the south NU sum we have

$$G_\theta(P_*) = W_{\eta_1} P_{*\eta_1} + W_{\eta_2} P_{*\eta_2} + \sum_{\nu_N} W_l^N P_{*l} - W_{\xi_1} P_{*\xi_1} - W_{\xi_2} P_{*\xi_2}$$

$$- \sum_{\nu_S} W_l^S P_{*l}$$

$$G_\theta(P_*T)_k = \frac{P_{*o} + P_{*\eta_1}}{2} \frac{T_o + T_{\eta_1}}{2} W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2} \frac{T_o + T_{\eta_2}}{2} W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2} \frac{T_o + T_l}{2} - \frac{P_{*o} + P_{*\xi_1}}{2} \frac{T_o + T_{\xi_1}}{2} W_{\xi_1}$$

$$- \frac{P_{*o} + P_{*\xi_2}}{2} \frac{T_o + T_{\xi_2}}{2} W_{\xi_2} - \sum_{\nu_S} W_l^S \frac{P_{*o} + P_{*l}}{2} \frac{T_o + T_l}{2}$$

$$L_\theta(P_*\phi)_k = \frac{P_{*o} + P_{*\eta_1}}{2} (\phi_{\eta_1} - \phi_o)_{k-\frac{1}{2}} W_{\eta_1}$$

$$+ \frac{P_{*o} + P_{*\eta_2}}{2} (\phi_{\eta_2} - \phi_o)_{k-\frac{1}{2}} W_{\eta_1} + \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2} (\phi_l - \phi_o)_{k-\frac{1}{2}}$$

$$- \frac{P_{*o} + P_{*\xi_1}}{2} (\phi_{\xi_1} - \phi_o)_{k-\frac{1}{2}} W_{\xi_1} - \frac{P_{*o} + P_{*\xi_2}}{2} (\phi_{\xi_2} - \phi_o)_{k-\frac{1}{2}} W_{\xi_2}$$

$$- \sum_{\nu_S} W_l^S \frac{P_{*o} + P_{*l}}{2} (\phi_l - \phi_o)_{k-\frac{1}{2}}$$

$$H(K) = W_J^{EW} \left\{ \frac{P_{*o} + P_{*E}}{2} \frac{u_o + u_E}{2} - \frac{P_{*o} + P_{*W}}{2} \frac{u_o + u_W}{2} \right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2} \frac{v_o + v_{\eta_1}}{2} W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2} \frac{v_o + v_{\eta_2}}{2} W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2} \frac{v_o + v_l}{2} - \frac{P_{*o} + P_{*\xi_1}}{2} \frac{v_o + v_{\xi_1}}{2} W_{\xi_1}$$

$$- \frac{P_{*o} + P_{*\xi_2}}{2} \frac{v_o + v_{\xi_2}}{2} W_{\xi_2} - \sum_{\nu_S} W_l^S \frac{P_{*o} + P_{*l}}{2} \frac{v_o + v_l}{2}$$

$$D3(M,K) = W_J^{EW} \left\{ \frac{P_{*o} + P_{*E}}{2} \frac{u_o + u_E}{2} \frac{\psi_o + \psi_E}{2} - \frac{P_{*o} + P_{*W}}{2} \frac{u_o + u_W}{2} \frac{\psi_o + \psi_W}{2} \right\}$$

$$+ \frac{P_{*o} + P_{*\eta_1}}{2} \frac{v_o + v_{\eta_1}}{2} \frac{\psi_o + \psi_{\eta_1}}{2} W_{\eta_1} + \frac{P_{*o} + P_{*\eta_2}}{2} \frac{v_o + v_{\eta_2}}{2} \frac{\psi_o + \psi_{\eta_2}}{2} W_{\eta_2}$$

$$+ \sum_{\nu_N} W_l^N \frac{P_{*o} + P_{*l}}{2} \frac{v_o + v_l}{2} \frac{\psi_o + \psi_l}{2} - \frac{P_{*o} + P_{*\xi_1}}{2} \frac{v_o + v_{\xi_1}}{2} \frac{\psi_o + \psi_{\xi_1}}{2} W_{\xi_1}$$

$$- \frac{P_{*o} + P_{*\xi 2}}{2} \frac{v_o + v_{\xi 2}}{2} \frac{\psi_o + \psi_{\xi 2}}{2} W_{\xi 2} - \sum_{\nu S} W_l^S \frac{P_{*o} + P_{*l}}{2} \frac{v_o + v_l}{2} \frac{\psi_o + \psi_l}{2}$$

PRT = PJ(1, K1, I) (GRHS.258)

$PRT = P_{*o}$

GTP = .5 * (GTP - WNSJ * PRT) (GRHS.260)

$$G_\theta (P_*) = \tfrac{1}{2}\{ W_{\eta 1} P_{*\eta 1} + W_{\eta 2} P_{*\eta 1} + \sum_{\nu N} W_l^N P_{*l} - W_{\xi 1} P_{*\xi 1} - W_{\xi 2} P_{*\xi 2} - \sum_{\nu S} W_l^S P_{*l}$$

$$- P_{*o}(W_N - W_S)\}$$

From 2.8 KH

$$G\theta(P_*) = (\Sigma_N - \Sigma_S) \frac{P_{*l} + P_{*o}}{2} W_l - P_{*o}(W_N - W_S)$$

$$= \tfrac{1}{2} (\Sigma_N - \Sigma_S) \{ P_{*l} W_l \} + \tfrac{1}{2} P_{*o} (\Sigma_N - \Sigma_S) W_l - P_{*o}(W_N - W_S)$$

$$= \tfrac{1}{2} (\Sigma_N - \Sigma_S) \{ P_{*l} W_l \} + P_{*o}\left\{ \frac{W_N - W_S}{2} - (W_N - W_S) \right\}$$

$$= \tfrac{1}{2} (\Sigma_N - \Sigma_S) \{ P^*_l W_l \} - P_{*o}(W_N - W_S) \text{ as above.}$$

GTPT(K) = GTPT(K) - WNSJ * PRT * PJ(3,k,I) (GRHS.263)

$$G\,(P_*T)_k = (\Sigma - \Sigma) \frac{P_{*o} + P_{*l}}{2} \frac{T_o + T_l}{2} W_l - (\Sigma_N - \Sigma_S) P_{*o} T_o$$

H (1) = H(1) * SIG1 (1) GRHS.266

$H_1(1) = H_1(1)\ \Delta k\sigma$

IF (KV .EQ. 1) GO TO 43

Skips loop if only one level.

H(K) = H(K-1) + H(K) * SIG1(K)

$$H_1(K) = \sum_{k'=1}^{k} H_1(K')\ \Delta k\sigma \quad (3.6A\ KH)$$

RAP = -H(KV) (GRHS.271)

$$\frac{\partial P_{*o}}{\partial t} = \sum_{k'=1}^{kv} H_1 \ \Delta_{k'}\sigma \qquad (3.5A\ KH)$$

IF (KV .EQ. 1) GO TO 52 (GRHS.273)

If KV = 1, then $\bar{\omega}$ array is complete, and equal zero at top and bottom of atmosphere.

ZB(K) = -RAP * SIG(K) - H(K-1)

Notice that ZB(K) = $P_{*o}\,\overline{\omega}_{k-\frac{1}{2}}$

$$P_{*o}\,\overline{\omega}_{k-\frac{1}{2}} = \sigma_{k-\frac{1}{2}} \sum_{k'=1}^{kv} H_l\,\Delta_{k'}\sigma - \sum_{k'=1}^{k} H_l\,\Delta_{k'}\sigma \qquad (3.6A \quad KH)$$

Z(K) = .5 * (ZB(K+1) + ZB(K)) + SIG3(K) * (RAP+GLP*PJ(1,K,I)) + GTP * PJ(2,K,J)

$$\omega_k = P_{*o}\frac{(\overline{\omega}_{k+\frac{1}{2}} + \overline{\omega}_{k-\frac{1}{2}})}{2} + \frac{\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}}}{2}\left[\frac{\partial P_{*o}}{\partial t} + u_o G_\lambda (P_*)_k + v_o G_\theta (P_*)_k\right] \qquad (3.7A \quad KH)$$

IF (KV - 2) 53, 47, 48

Since $\overline{\omega}_{kv+\frac{1}{2}} = \overline{\omega}_{\frac{1}{2}} = 0$, $ZB2_{KV}$ = SIG2(KV) * ZB(KV1) = 0, and $ZB1_1$ = SIG2(1) * ZB(1) = 0. If KV = 1, then $ZB1_{KV} = ZB2_1 = 0$, so that the extra term (2.4 KH) does not appear in the divergence.

If KV = 2, then KV - ½ ≠ ½; although in this case then $ZB1_{KV} \neq 0$ and $ZB2_1 \neq 0$; so that

$$D3(M,KV) = (\sum_E - \sum_W)\left\{\frac{P_{*o} + P_{*l}}{2}\,\frac{u_o + u_l}{2}\,\frac{\psi_o + \psi_l}{2}\,W_l\right\}$$
$$+ (\sum_N - \sum_S)\left\{\frac{P_{*o} + P_{*l}}{2}\,\frac{v_o + v_l}{2}\,\frac{\psi_o + \psi_l}{2}\,W_l\right\}$$
$$- \frac{P_{*o}\,\omega_{kv-\frac{1}{2}}}{\sigma_{kv+\frac{1}{2}} - \sigma_{kv-\frac{1}{2}}}\,\frac{\psi_{kv} + \psi_{kv-1}}{2}$$

where

$$Y_{kv-\frac{1}{2}} = \frac{\psi_{kv} + \psi_{kv-1}}{2} \qquad (2.4 \quad KH)$$

and

$$D3(M,1) = (\sum_E - \sum_W)\left\{\frac{P_{*o} + P_{*l}}{2}\,\frac{u_o + u_l}{2}\,\frac{\psi_o + \psi_l}{2}\,W_l\right\}$$
$$+ (\sum_N - \sum_S)\left\{\frac{P_{*o} + P_{*l}}{2}\,\frac{v_o + v_l}{2}\,\frac{\psi_o + \psi_l}{2}\,W_l\right\}$$
$$+ \frac{P_{*o}\,\overline{\omega}_{3/2}}{\sigma_{3/2} - \sigma_{1/2}}\,\frac{\psi_1 + \psi_2}{2}$$

where

$$Y_{3/2} = \frac{\psi_1 + \psi_2}{2}$$

If KV > 2, then both terms ZB1 and ZB2 are present simultaneously, for indices K = 2,...,KVM, and

$$D3(M,K) = (\sum_E - \sum_W)\left\{\frac{P_{*o} + P_{*1}}{2}\,\frac{u_o + u_1}{2}\,\frac{\psi_o + \psi_1}{2}\,W_1\right\}$$

$$+ (\sum_N - \sum_S)\left\{\frac{P_{*o} + P_{*1}}{2}\,\frac{v_o + v_1}{2}\,\frac{\psi_o + \psi_1}{2}\,W_1\right\}$$

$$+ \frac{P_{*o}}{\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}}\left\{\bar{\omega}_{k+\frac{1}{2}}\,\frac{\psi_{k+1} + \psi_k}{2} - \bar{\omega}_{k-\frac{1}{2}}\,\frac{\psi_k + \psi_{k-1}}{2}\right\}$$

CALL FRC
CALL HEAT (GRHS.304)

Forcing function subprogram

RTE(1,K) = -D3(1,K) + T1 * PJ(2,K,I) - R * GLPT(K) - T2SIG2(K) + FID(1,K)

$$\frac{\partial(P_{*o}u_o)}{\partial t} = -D\left[\frac{u_o + u_1}{2}\right] + P_{*o}\left\{2\Omega\sin\theta_j + \frac{\tan\theta_j}{a}\,u_o\right\}v_o - RG_\lambda(P_*T)$$

$$- \frac{\sigma_{k+\frac{1}{2}}L_\lambda(P_*\phi)_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}L_\lambda(P_*\phi)_{k-\frac{1}{2}}}{2(\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}})} + (F_\lambda)_o \qquad \text{(3.1A KH)}$$

Note: The extra factor of 2 in the denominator of the L_λ term is misleading. L_λ, as calculated above, needs to absorb factor ½ to agree with L_λ in KH.

RTE(2,K) = -D3(2,K) - T1 * PJ(1,K,I) - R * GTPT(K) - T3 * SIG2(K) + FID(2,K)

$$\frac{\partial P_{*o}v_o}{\partial t} = -D\left[\frac{v_o + v_1}{2}\right] - P_{*o}\left\{2\Omega\sin\theta_j + \frac{\tan\theta_j}{a}\,u_o\right\}u_o - RG_\theta(P_*T)$$

$$- \frac{\sigma_{k+\frac{1}{2}}L_\theta(P_*\phi)_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}}L_\theta(P_*\phi)_{k-\frac{1}{2}}}{2(\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}})} + (F_\theta)_o \qquad \text{(3.2A KH)}$$

RTE(3,K) = -D3(3,K) + SIG5(K) * Z(K) * PJ(3,K,I) + CPRT * QDT(K) + FID(3,K)

$$\frac{\partial(P_{*o}T_o)}{\partial t} = -D\left[\frac{T_o + T_1}{2}\right] + \frac{R}{c_p}\,\frac{\omega_k T_o}{\frac{\frac{1}{2}(\sigma_{k+\frac{1}{2}} + \sigma_{k-\frac{1}{2}})}{2}} + \frac{P_{*o}}{c_p}(\dot{q})_o + (F_T)_o$$

(3.3A KH)

RTE(4,K) = -D3(4,K) - PRT * CORA(K) + FID(4,K)

$$\frac{\partial P_{*o}}{\partial t} = -D\left[\frac{\tau_o + \tau_1}{2}\right] - P_{*o}(CON)_o + (F_\tau)_o$$

SUBROUTINE YSL(PJ, PJL)

This routine returns Y at the coincident boundary between cells L and I at all KV levels. The version here described is the filtered factor form; other versions of YSL have been programmed, and are available through distinct update deck names (GYSL2, GYSL3).

Common Blocks

/FC/

/BUF/

/UNITS/

/AC/

/CEFG/

Variable Description

K Level index for do loop.

M Index for 3-D prognostic variable in YP array.

PRT $\frac{P_{*o} + P_{*l}}{8}$

YT $\frac{P_{*o} + P_{*l}}{2} \frac{Y_o + Y_l}{2}$ where Y is u(MB = 1) or v(MB = 2).

Helpful Comments

1 CONTINUE (GYSL.39)

YSL returns the arrays Y1 and YP to RHS. For integration along the east or west boundary (MB = 1), Y1 holds $\frac{P_{*o} + P_{*l}}{2} \frac{u_o + u_l}{2}$ and YP holds $\frac{P_{*o} + P_{*l}}{2} \frac{u_o + u_l}{2} \frac{\psi_o + \psi_l}{2}$ at all KV levels, where ψ is a 3-D variable. For integration along the north or south boundary (MB = 2), Y1 holds $\frac{P_{*o} + P_{*l}}{2}$ $\frac{v_o + v_l}{2}$ and YP holds $\frac{P_{*o} + P_{*l}}{2} \frac{v_o + v_l}{2} \frac{\psi_o + \psi_l}{2}$ at all KV levels.

SUBROUTINE PST(PJ,PJL)

This routine returns PstarT at the coincident boundary between cells L and I, at all KV levels. Other versions of PST have distinct deck names (GPST2), the version here described is the filtered factor form.

Common Blocks

/FG/

/BUF/

/UNITS/

/AC/

/CEFG/

Variable Description

K Level index for do loop.

PRT $\dfrac{P_{*o} + P_{*l}}{4}$

Helpful Comments

1 CONTINUE (GPST1.20)

PST returns array PT to RHS. For integration along a cell boundary PT holds $\dfrac{P_{*o} + P_{*l}}{2} \dfrac{T_o + T_l}{2}$, where l is the zonal index of the contiguous cell.

SUBROUTINE PSP(PJ,PJL,PHJ,PHL)

This routine returns PstarPHI, at the coincident boundary between cells L and I, at all KV vertical levels. The version here described is the K and H form, other versions are available through distinct update deck names.

Common Blocks

/FG/

/BUF/

/UNITS/

/AC/

/CEFG/

Variable Description

K Level index for do loop.

PRT $\dfrac{P_{*o} + P_{*l}}{2}$

Helpful Comments

1 CONTINUE (GPSP1.37)

PSP returns the array PH to RHS. PH holds $\dfrac{P_{*o} + P_{*l}}{2} (\phi_{l_{k-\frac{1}{2}}} - \phi_{o_{k-\frac{1}{2}}})$ at all KV levels, where l is the zonal index of the contiguous cell on its band (JM1 for north, JP1 for south, J for east-west).

SUBROUTINE ENTIS

End time integration scheme. This routine completes necessary I/O at the end of a time step. INPUT file IN1 and IN2 are rewound, and output file IOT has records JB1 to JB5 written on it. IOT is then also rewound.

Common Blocks

/LTC/

/BUF/

/UNITS/

/AC/

Description of Variables

I Index for do loop.

NBJ1 Number of cells on last data band JB.

Helpful Comments

2 REWIND IN1 $ NBJ1 = NB(JB1) $ CALL CBUFT(IOT,0,99,U,PJI,12*NBJ1)

ENTIS is called from ROTATE directly after the call to TISϕ was processed with J = JB1. During this call TISϕ updated the last data band using

P3(J=JB), P4(J=JB1), and P5(J=JB1+1). This data was placed in array P2, which is passed by ROTATE to ENTIS as the array PJ1. Notice, also, the anticipatory buffering of P6(J=JB1+2=JB3) at the beginning of TISϕ. For EULER steps then the south polar dummy bands JB2, JB3 are required. For more general schemes, dummy bands JB2-JB5 are needed. The call to CBUFT writes PJ1 on the output file. For the single level case, PJ is dimensioned 4 x 3 x NBJX. The updated JB1 data is in 4 x 3 x NBJ1 = 12 x NBJ1. The 99 is a unique identifier for the BERROR routine, and the zero identifies the call within ENTIS.

```
DO 3 I = 1,3
3  CALL PBUFT(IOT,I,99, U, BLK,3) $ CALL CBUFT(IOT,4,99,U,BLK,3)
```

South polar dummy bands JB2-JB5 are output on file IOT. PBUFT performs a unit check before the buffer, but not after, and thus avoids two initialization times. I, again, identifies the call to PBUFT within the loop.

```
IF (IDX .EQ. 1) END FILE IDES $ RETURN
```

All active data files are now rewound. If a display file was created in this ts., an EOF is written. AU is now ready for generating the next ts.

APPENDIX C: DISPLAY

PROGRAM DISPLY

Displays files saved from GEX run.

Characteristics

Core: 125000_B

Time: Open

Options: Described below

Organization of Core

	/BUF/	101	14600
	/UNITS/	14701	15
	/AC/	14716	2260
Display Options	REPORT	17176	657
	DISPLY	31451	25564
INITIALIZATION PROGRAMS	SKFL	57235	220
	GEN	57455	35
SYSTEM			
	/ /	102737	12000

Common Blocks

/SPLNCO/ Parameters used in spline option.

/PLC/ Axis labels and parameters for plots.

/BUF/ Used to expand a display file.

/UNITS/

/AC/

/DPC/ Parameters used in spectral plotting.

Description of Variables

AM Array of second derivatives (moments) returned by spline fitting subroutine.

AXLAB Array of BCD characters used to label x axis.

AYLAB Array of BCD characters used to label y axis.

BLD Array of values of current variable at grid points. BLD is a mirror image of FLD, since the CONTOR subroutine expects Y values in increasing order.

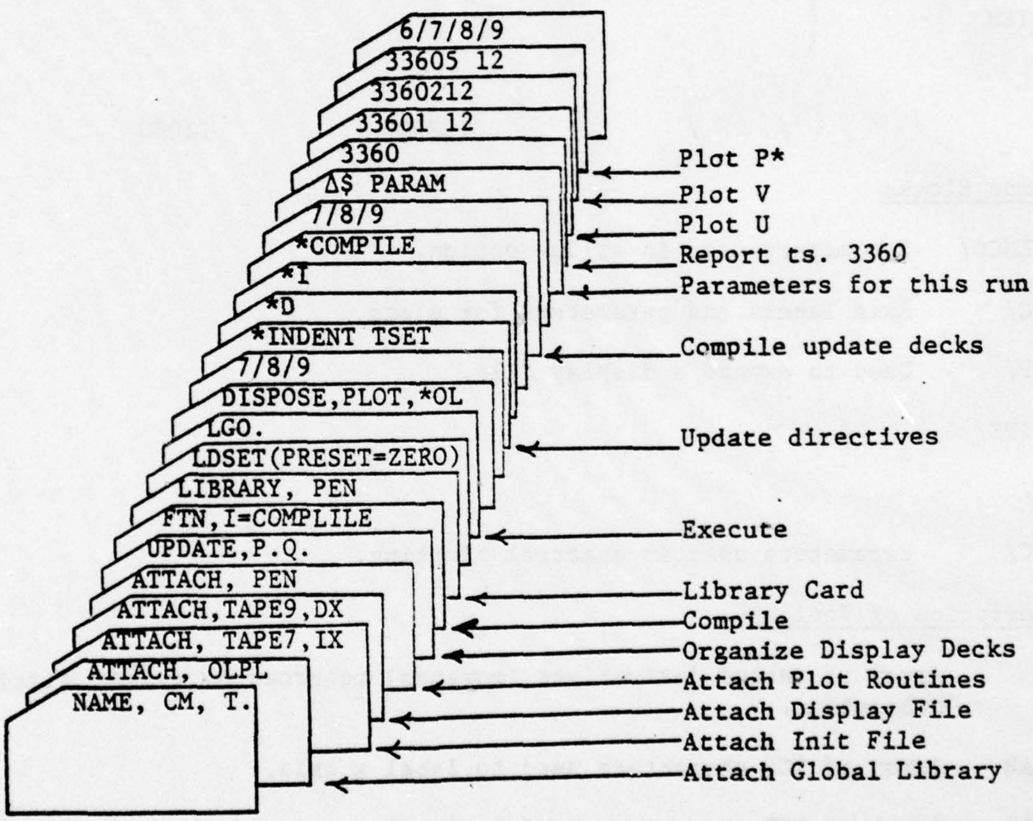
CZ Array of spectral components returned by FFT routine.

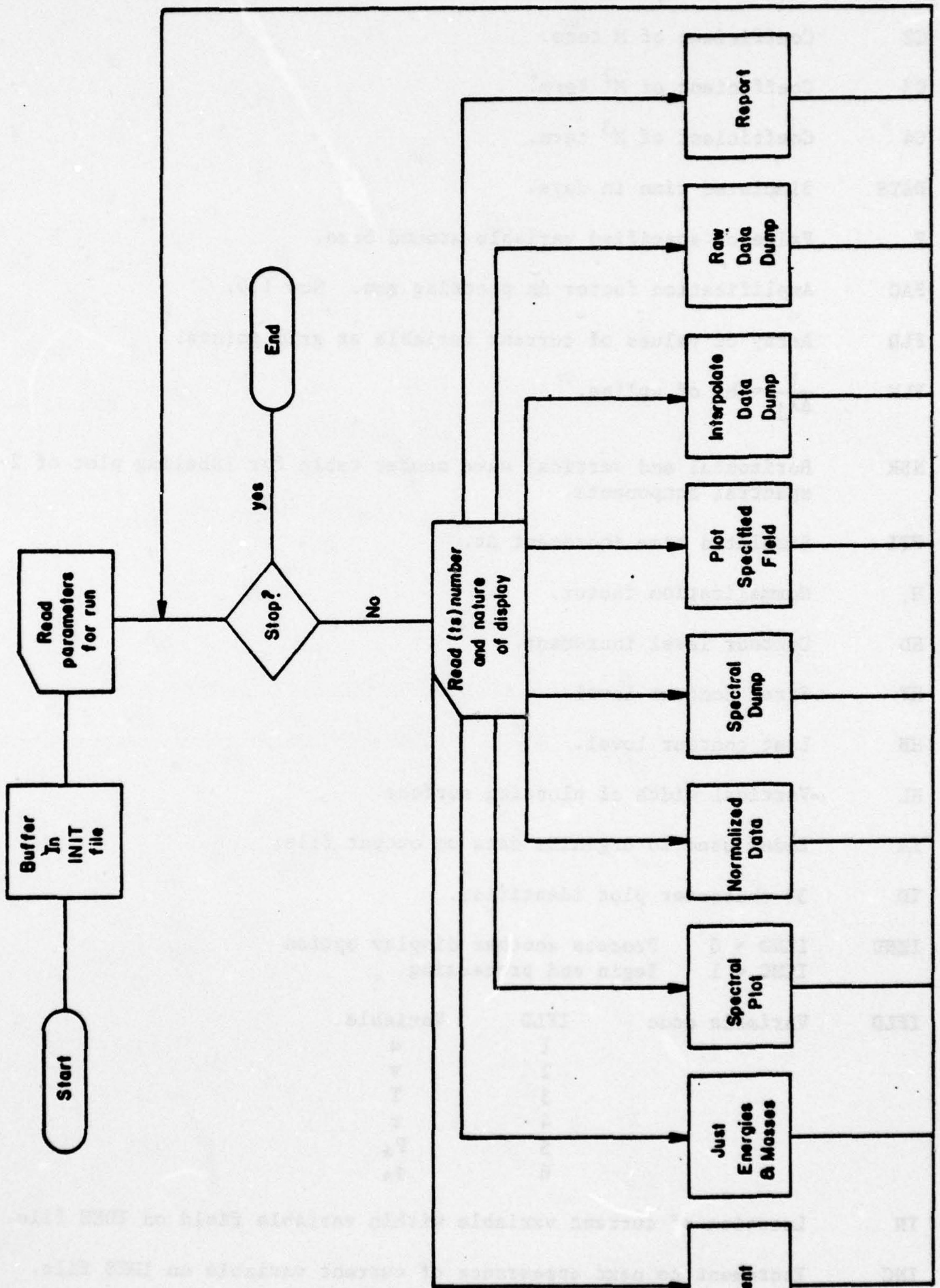
C1 Coefficient of constant term in spline.

FILES

INPUT	(5)	Community Input File
OUTPUT	(6)	Community Output File
INIT	(7)	Private Binary File of Grid Information
ISTAR1	(8)	Scratch File
IDES	(9)	Display File

Display Deck (1 possibility)




Start
Buffer In INIT file
Read parameters for run
Stop?
yes
End
No
Read (is) number and nature of display
Ident
Just Energies & Masses
Spectral Plot
Normalized Data
Spectral Dump
Plot Specified Field
Interpolate Data Dump
Raw Data Dump
Report

C2	Coefficient of M term.
C3	Coefficient of M^2 term.
C4	Coefficient of M^3 term.
DAYS	Simulated time in days.
F	Value of specified variable around band.
FAC	Amplification factor in plotting run. Now 1.0.
FLD	Array of values of current variable at grid points.
FLN	$\frac{1}{\Delta\lambda_j}$ = hi of spline.
NBR	Horizontal and vertical wave number table for labeling plot of 2-D spectral conponents.
FTI	Simulated time increment Δt.
H	Normalization factor.
HD	Contour level increment.
HF	First contour level.
HH	Last contour level.
HL	Vertical width of plotting surface.
IA	Index used to organize data on output file.
ID	30-character plot identifier.
IEND	IEND = 0 Process another display option IEND = 1 Begin end processing
IFLD	Variable code

IFLD	Variable
1	u
2	v
3	T
4	r
5	P_*
6	ϕ_*

IN	Location of current variable within variable field on IDES file.
INC	Increment to next appearance of current variable on IDES file.
INTRP	INTRP = 0 Spline option not called INTRP = 1 Spline routines are called

IPLT	IPLT = 0	Run does no plotting
	IPLT = 1	Plotting subroutines are called

ISC Scratch file name (not used).

I1 Counter used to organize data for dumps.

I1P I1 + 1

I2 Counter used to organize data for dumps.

I2P I2 + 1

JB Total number of latitude bands.

JP Wave number J - 1.

$\left.\begin{array}{l}\text{J1}\\ \text{J2}\\ \text{J3}\\ \text{J4}\end{array}\right\}$ Loop controls.

L Array index.

LEV Vertical level number of current variable.

LT Index for stripping BLK of current variable

LTS Current time step displayed.

LTSL Previous time step displayed, or -1 if no previous step.

LZ Number of words in a data block in IDES file.

N Loop index.

NBJ b_j

NBJM $b_j - 1$

NCL Number of contour lines in a plot.

NLCX Number of BCD characters in label for horizontal axis.

NLCY Number of BCD characters in label for vertical axis.

NP Number of pages to accomodate data dump.

NPI Loop index.

NPX	Loop index.	
NT	2 * JB	
NTY	Defines plot option.	
	NTY = 0 and IFLD = 0	Report
	NTY = 0	Raw data dump of field IFLD
	NTY = 1	Interpolated data dump of field
	NTY = 2	Plot field IFLD
	NTY = 3	Spectral dump of field IFLD
	NTY = 4	Normalized spectral dump of field
	NTY = 5	Spectral plot of field IFLD
NUMB	Array of numbers used to label printed output from dumps.	
N1	N2 + 1	
N2	½NBJX	
N2P	N1	
N3	Loop index.	
RFLN	$\Delta\lambda_j$	
T	Output string from contour plot.	
X	Zonal grid table.	
XL	Maximum length of plotting surface.	
XNP	Fractional number of pages for dump output.	
XNT	Truncated number of output pages.	
XT	Temporary array of interpolated values.	
Y	Meridional table in degrees.	
YT	Temporary array of interpolated values.	
ZMAX	Maximum of current variable over globe.	
ZMIN	Minimum of current variable over globe.	

Helpful Comments

ISTAR1 = ISC

ISTAR1 = 8 is used as a scratch file to expand compressed data file for report output.

CALL CBUFN(INIT,0,0,U,CBLK,1200) (GSPLAY.25)

Fills out AC common with grid information for display file.

CALL GEN(X,Y) $ IF (INTRP. EQ. 1) CALL SPREP(NBJX)

GEN returns arrays X, Y; $x_i = (i-1) \min_j \{\Delta\lambda_j\}$ and $y_j = \theta_{JB1-j}$ measured in degrees. SPREP intitalizes values for the cubic splines routine. NBJX has no significance in the call here; and is not used in SPREP.

AXLAB(Y) = SH_DAYS (GSPLAY.30)

Axis labels for contour plots.

INC = 2 + MP * (K1-1)

Determines number of variable position to pass over on compressed data file.

LTS = LTSL

Initializes last time step accessed on display file to -1.

699 IF (LTS. GE. 0) LTSL = LTS

If a display file was previously accessed, save its ts. in LTSL. This value will be used to seek the next file (LTS) in the most economical way. All display options will loop back to 699 for a decision on further processing.

IF (IEND. EQ. 0) GO TO 6999 $ IF (IPLT .EQ. 1) CALL ENDPLT $ STOP

Processing ends if the previous read statement set the code IEND = 1. A call to ENDPLT is mandatory if this run created any plot files.

6999 READ(INPUT,700) LTS, IFLD, LEV,NTY, IEND

Time step LTS is to be accessed on the IDES file and displayed. The nature of the display is determined by the values read. See sample deck.

DAYS = LTS * FTI/86400.

Simulated time of ts. LTS in days.

ENCODE(10,502) AXLAB(3)) DAYS $ CALL SKFL(IDES,LTS,LTSL)

These characterizing numbers are reformatted in internal BCD code, to be

used as axis labels if plotting is indicated. The call to SKFL locates the display file labeled LTS, and positions that file directly after the 3-word label.

<u>IF ((NTY .NE. 0) .OR. (IFLD .NE. 0)) GO TO 698 $ BACKSPACE IDES</u>

The code for report output has both NTY = IFLD = 0. In this case the IDES file LTS must be first expanded onto a scratch file (ISTAR1), before it is sent to REPORT. Since EXPAND expects a file positioned at the beginning of the 3-word label, the backspace over 1 record is done.

<u>GO TO (1000, 2000, 3000, 4000, 5000, 6000) IFLD</u>

The axis labels are now ready

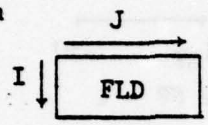
	1	2	3	4
AXLAB	u, v, T, OZ, P_*, ϕ_*	ts #	days #	DAYS
AYLAB	DATA SER	serial #	GLOBAL RU	N

The value of IN locates a variable within a data block on a compressed file. P_* and ϕ_* occupy positions 1 and 2 respectively, then u, v, T, r follow consecutively up a column. IN locates one of these variables at a fixed level.

The logic of the loop is:

```
——N3 = 1, NBLK
| LT = IN            Locate first occurrence of variable within present
|                    data block at level to be ripped off.
| J1 = J2 + 1        First band is present block
| J2 = JLI(N3)       Last band is present block
| LZ = LSZ(N3)       Number of words in present block
|                Buffer in data block
| BUF(LZ) <——————————————————————————————    IDES
|                    N3
|——J = J1, J2
| NBJ = NB(J)        Number of cells in present band
| L = J - 1          Latitude index in FLD array
|——I = 1, NBJ
| FLD(I,L) = BLK(LT)   Rip off variable within present data block.
| LT = LT + INC      Next occurrence of variable at same level is separated
|                    by INC words.
——6002
```

Notice that before entering the loop IDES is positioned after the 3-word label for ts. LTS. On leaving the loop, IDES is positioned before the EOF of ts. LTS, and, FLD contains a full globe of the specified variable at a fixes level, in the form



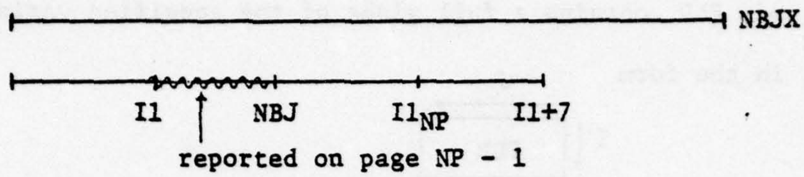
IF (NTY .NE. 0) GO TO 6003 $ XNP = FLOAT(NBJX)/8.0 $ NP = XNP $ XNT = NP

NTY = 0 produces a raw data dump of the field held in FLD. The printed output should write the two dimensional array with a flexible number of cells on each band. The data is to be reported horizontally by zonal index, and, vertically by latitude index. To print page numbers, the data is partitioned by determining the number of pages needed to report the band with the largest number of cells, eight cells to a page (XNP). This value is truncated (NP), and then floated (XNP). If XNP > XNT, then reporting 8 cells to a page will require an extra fraction of a page to accomodate the band with NBJX cells (NP = NP + 1).

The logic of the loop is:

```
I1 = -7
NPX = 1, NP
I1 = J1 + 8        Smallest zonal index on this page
WRITE(IPRT,7001)     Write specified variable (labeling and num-
                     bering pages).
I2 = I1 + 7        Largest zonal index on this page
J = 1, JB
NB = NB(J+1)       Number of cells on data band J
WRITE NUMB(J1P)--NUMB(I2P)   NUMB(M+1) = M so this writes the
                                  zonal indices I1 thru I2 across page
I2 = MINφ(I1+7,NBJ)   There are a maximum of NBJ(<NBJX) cells to
                      be reported on this band
YES   I1 > I2?
         WRITE NUMB(J+1)              Write latitude index J followed
            FLD(I1,J) thru FLD(I2,J) by segment I1 thru I2 of variable
         GO TO 6004
      WRITE (IPRT,7003)    Skip record since all cells on this band have
                           been written
6004
```

The situation I1 > I2 can occur if data thru NBJ has been reported on page NP - 1 (so that I2 = NBJ). Then $I1_{NP} = I1_{NP-1} + 8$, so that



then I2 = MINϕ(I1+7,NBJ) < $I1_{NP}$, indicating data band j reported.

```
6003  IF (NTY .EQ. 1) GO TO 6011
      IF (NTY .GT. 2) GO TO 6020
```

NTY = 2 indicates a contour plot of the variable in FLD. Such plots can then be made level by level, for any specified variable. The array FLD is first flipped into BLD, so that column 1 of BLD contains data next to the south pole, and column JB contains data next to the north pole. Since the Y array returned by GEN is in increasing order ($\theta_S = -90^o \longrightarrow \theta_N = +90^o$), the variable field must be in the same order. That is, J = 1 should correspond to the grid point next to the south pole, and J = JB that next to the north pole, as in array BLD.

CALL CONTOR (BLD,66,34,X,Y,NBJX,JB,XT,YT,T,998,HF,HD,HH,NCL)

Produces contour plot of data field BLD.

CALL PLOT(13,0.,-3)

Moves pen, with point up, 13 inches to right of origin.

C INPERPOLATE TO REGULAR GRID

In this option, the data for the variable in FLD is interpolated, band by band, to a regular grid of NBJX data points per band. Cubic splines are used to replace the discrete data around a band, with continuous polynomials, on which the data is evaluated at the NBJX regular arguments.

6014 F(I) = FLD(I,J) $ CALL SPREP2(NBJ) $ CALL SPCOFP(NBJ)

F holds the NBJ grid values of the variable around the current band J. SPREP2 evaluates certain constants needed for the matrix elimination for this band, and SPCOFP produces the coefficients of the cubics fitting this data, along with the second derivatives at the grid points.

CALL SPINT(NBJX,DEV,FLD,NBJ,J) (GSPLAY.88)

Replaces J^{th} column of FLD with NBJX interpolated data values.

C INTERPOLATED DATA DUMP (GSPLAY.90)

The interpolated data is reported horizontally by zonal index, and vertically by latitude index. Pages are numbered as described previously. There is no need for the test I1 > I2, since all columns of FLD are of length NBJX.

C DOUBLE PERIODIC FOURIER TRANSFORM (GSPLAY.98)

Spectral data options follow.

6020 CALL DPFT

Returns 2-dimensional spectral components in FLD.

6036 WRITE IPRT,712) JP, (FLD(I,J),I=I1P,I2P) $ IF (NTY .EQ. 3) GO TO 699

The loop above this point points the 2-dimensional spectral components of the field FLD, numbering pages, and wave number components horizontally and vertically.

6045 CONTINUE $ IF (NTY .EQ. 4) GO TO 699

The loop above this points the 2-dimensional normalized spectral components of the field FLD, numbering pages and components.

6060 NLCX = 50 $ CALL CONTOR(FLD,66,FNBR,FNBR,NLJB1XT,YT,T,998,HF,HD,HH,NCL) $
NLCX = 40 $ CALL PLOT (13,0.,-3) $ GO TO 699

Plots field of 2-dimensional spectral components normalized above.

6050 BACKSPACE IDES $ REWIND ISTAR1 $ CALL EXPAND(IDES,ISTAR1)
CALL REPORT(ISC,-1) $ GO TO 699

The -1 codes REPORT to just print the energies and masses for the specified time step.

SUBROUTINE GEN(X,Y)

Returns to DISPLY arrays X, Y. These are the zonal and meridional tables passed to the plotting routines, if that option is chosen in DISPLY.

Common Blocks

/BUF/

/UNITS/

/AC/

Variable Description

DX Minimal zonal increment in degrees $\Delta\lambda_{j_x}$.

J Data band index.

JJ Reverse index.

RTD ½ conversion factor from radians to degrees $\frac{1}{2}\left(\frac{180}{\pi}\right)$

Helpful Comments

10 X(I) = X(I-1) + DX

$X(1) = 0, \ldots, X(K) = (K-1)\Delta\lambda_{j_x}, \ldots, X(NBJ) = N(BJX-1)\Delta\lambda_{j_x}$

Thus $X(NBJX+1) \equiv X(1) = 360^{\circ}$

11 Y(5) = RTD * (BAND(JJ+3) + BAND(JJ+2))

$Y(1) = RTD * (BAND(JB2) + BAND(JB1)) = RTD * (\theta_{JB+\frac{1}{2}} + \theta_{JB-\frac{1}{2}}) = \theta_{JB}$

.
.
.

$Y(JB) = RTD * (\theta_{3/2} + \theta_{1/2}) = \theta_1$

Then $Y(\frac{1}{2}) \equiv \theta_{JB+\frac{1}{2}}$ is the south pole (-90°) and $Y(JB+\frac{1}{2}) \equiv \theta_{\frac{1}{2}}$ is the north pole ($+90^{\circ}$). Note GEN returns X and Y in increasing order for CONTOR.

SUBROUTINE SKFL(IU,L,LTSL)

Subroutine SKFL is called from DISPLY to seek the file labeled with the ts. L, on the tape IU. On entry, IU is positioned either at the beginning of information (LTSL = -1), or, at the end of the data block of the time step (LTSL) previously found by SKFL.

Common Blocks

/BUF/ ———

/UNITS/

/AC/

Variable Description

N Do loop index and code passed to CBUFN routine for locating source of buffer error.

NBL Value set if IU has to be backspaced to locate L (LTS < LTSL)

Helpful Comments

IF (LTS .LT. 0) GO TO 21 $ NBL = 2 * (NBLK + 1) + 1 (GSPLAY.141)

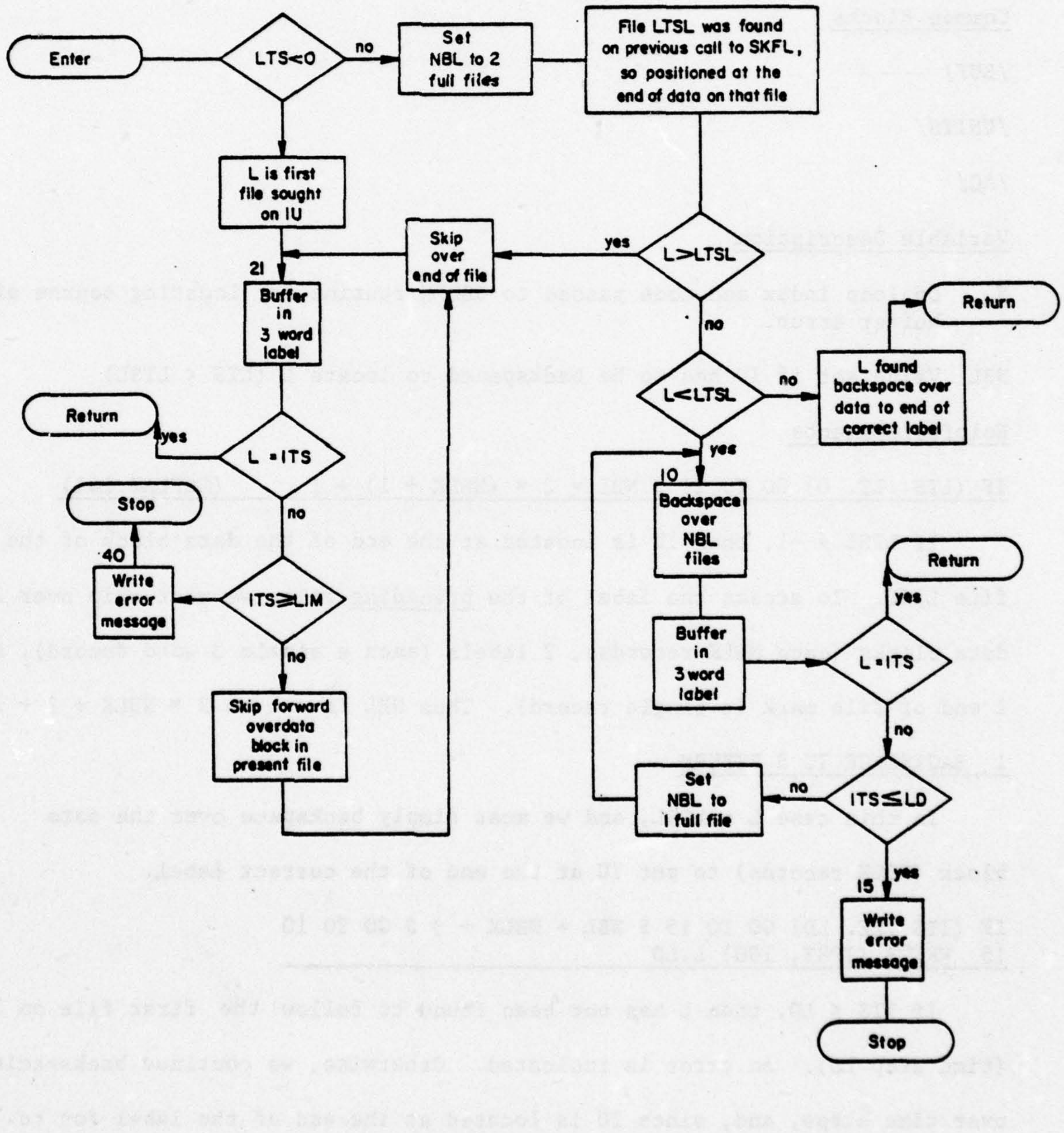
If LTSL ≠ -1, then IU is located at the end of the data block of the file LTSL. To access the label of the preceding file, we must skip over 2 data blocks (each NBLK records), 2 labels (each a single 3 word record), and 1 end of file mark (a single record). Thus NBL is set to 2 * NBLK + 2 + 1.

1 BACKSPACE IU $ RETURN

In this case L = LTSL, and we must simply backspace over the data block (NBLK records) to put IU at the end of the correct label.

IF (ITS .LE. LD) GO TO 15 $ NBL = NBLK + 3 $ GO TO 10
15 WRITE (IPRT, 100) L,LD

If ITS ≤ LD, then L has not been found to follow the first file on IU (time step LD). An error is indicated. Otherwise, we continue backspacing over time steps, and, since IU is located at the end of the label for ts. ITS, to access the label (of the preceding file we must skip over 1 data



Enter
LTS<0
no
Set NBL to 2 full files
File LTSL was found on previous call to SKFL, so positioned at the end of data on that file
L is first file sought on IU
Skip over end of file
yes
L>LTSL
21
Buffer in 3 word label
no
Return
no
L<LTSL
L found backspace over data to end of correct label
Return
yes
L = ITS
yes
10
Stop
no
Backspace over NBL files
40
Return
Write error message
ITS≥LIM
yes
Buffer in 3 word label
L = ITS
no
no
Skip forward overdata block in present file
Set NBL to full file
no
ITS≤LD
yes
15
Write error message
Stop

block (NBLK records), 2 labels (each a single 3 word record), and 1 end of file mark (a single record). Thus NBL = NBLK + 2 + 1.

```
IF (ITS .GE. LIM) GO TO 40 $ DO 3 N = 1, NBLK
3  CALL CBUFN(IU,I,N,U,BLK,3) $ GO TO 20             (GSPLAY.153)
```

If ITS $\geq$ LIM, then L has not been found to precede the last file on IU (time step LIM). An error is indicated. Otherwise, one data block (NBLK records) is skipped over to prepare to access the label of the file following ITS on IU.

SUBROUTINE SPREP(N)

This routine returns to DISPLY, through SPLNCO common, the coefficients of a cubic spline fit to the data in variable F, and, also, the second derivatives at the fitted points. F holds a full band of NBJ data points in SPLNCO common. The procedure is described in ANW. In the notation found there, the spline is

$$\text{A.}\quad S_{\Delta(x)} = M_{i-1}\frac{(x_i - x)^3}{6hi} + M_i\frac{(x - x_{i-1})^3}{6hi} + \left(y_{i-1} - \frac{M_{i-1}hi^2}{6}\right)\left(\frac{x_i - x}{hi}\right) + \left(y_i - \frac{M_i hi^2}{6}\right)\left(\frac{x - x_i - 1}{6}\right)$$

where, M_i is the second derivative·

$y_i = F_i$

hi is the grid spacing, here $\Delta\lambda_j$

For the periodic case: $M_1 = M_{N+1}$

$y_1 = y_{N+1}$

For equal spacing: $\dfrac{h_i}{h_i + h_{i+1}} = \dfrac{h_{i+1}}{h_i + h_{i+1}} = \dfrac{\Delta\lambda_j}{2\Delta\lambda_j} = \tfrac{1}{2}$

so that the defining equations for the second derivatives, M_i, are:

$$\begin{bmatrix} D_1 & \lambda & & & \mu \\ u & D_1 & \lambda & & \\ & \cdot & \cdot & & \\ & & \cdot\mu & D_1 & \lambda \\ \lambda & & & \mu & D_1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \cdot \\ \cdot \\ \cdot \\ M_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{bmatrix}$$

where $\lambda = \mu = \frac{1}{2}$, $D_1 = 2$

and;

$$\text{B.} \quad d_i = \frac{6}{2\Delta\lambda_j} \frac{1}{\Delta\lambda_j} \{y_{i+1} - 2y_i + y_{i-1}\} = \frac{3}{\Delta\lambda_j{}^2} \{y_{i+1} + y_{i-1} - 2y_i\}$$

The procedure is a variation of Gaussian elimination, taking advantage of the zero's. Do not try to relate the notation for this procedure in the program, to that found in the Ahlberg book. I emphasize, that you will be very confused if you do so.

The first step is to eliminate between the first and last rows.

$$\begin{bmatrix} D_1 & \lambda & & & Q_1 \\ u & D_1 & \lambda & & \\ & & & & \\ & & & & \\ \square - P_1\lambda & & \ldots & & \mu(D_1 - PIQ_1) \end{bmatrix} \begin{bmatrix} M_1 \\ \cdot \\ \cdot \\ \cdot \\ M_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \cdot \\ \cdot \\ \cdot \\ d_N - P_1 d_1 \end{bmatrix}$$

with $D_1 = 2$, $P_1 = \lambda/D_1$, $Q_1 = \mu$, $PIQ_1 = P_1Q_1$. Then eliminating between the first and second equations:

$$\begin{bmatrix} D_1 & \lambda & & & Q_1 \\ & D_2 & \lambda & & Q_2 \\ & & & & \\ & & & & \\ & -P_1\lambda & & \ldots & \mu(D_1 - PIQ_1) \end{bmatrix} \begin{bmatrix} M_1 \\ \cdot \\ \cdot \\ \cdot \\ M_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 - A_2 d_1 \\ \cdot \\ \cdot \\ d_N - P_1 d_1 \end{bmatrix}$$

with $A_2 = \mu/D_1$, $D_2 = D_1 - A_2\lambda$, $Q_2 = -A_2Q_1$, and (below) $P_2 = -P_1\lambda/D_2$, $PIQ_2 = PIQ_1 - P_2Q_2$.

Then eliminating between the second and last:

$$\begin{bmatrix} D_1 & \lambda & & Q_1 \\ & D_2 & \lambda & Q_2 \\ & & & \\ & & & \\ & & -P_2\lambda & \mu PIQ_2 \end{bmatrix} \begin{bmatrix} M_1 \\ \\ \\ \\ M_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 - A_2 d_1 \\ \\ \\ d_N - P_1 d_1 - P_2(d_2 - A_2 d_1) \end{bmatrix}$$

and so on until:

$$
\begin{bmatrix}
D_1 & \lambda & & & & Q_1 \\
& D_2 & \lambda & & & Q_2 \\
& & \ddots & & & \vdots \\
& & & D_{N-2} & \lambda & Q_{n-2} \\
& & & & D_{N-1} & (\lambda+Q_{N-1}) \\
& & & & (-P_{N-2}\lambda+\mu) & H_N
\end{bmatrix}
\begin{bmatrix} M_1 \\ \cdot \\ \cdot \\ \cdot \\ M_N \end{bmatrix}
=
\begin{bmatrix} \delta_1 \\ \delta_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta_{N-1} \\ d_N - P_1 d_1 - P_2(d_2 - A_2 d_1) - P_3(\ldots) \end{bmatrix}
$$

where $H_N = D_1 - \sum_{k=1}^{N-2} (PIQ)_k$, $\delta_i = (d_i - A_i(d_{i-1} - A_{i-1} \ldots)$ or, finally, eliminating between the last two equations using $\dfrac{P_{N-2}\lambda - \mu}{D_{N-1}} = -P_{N-1} - A_N$

$$
\begin{bmatrix}
D_1 & \lambda & & & & Q_1 \\
& D_2 & \lambda & & & Q_2 \\
& & \ddots & & & \vdots \\
& & & D_{N-2} & \lambda & Q_{N-2} \\
& & & & P_{N-1} & (\lambda+Q_{N-1}) \\
& & & & & \overline{D}_N
\end{bmatrix}
\begin{bmatrix} M_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ M_N \end{bmatrix}
=
\begin{bmatrix} \delta_1 \\ \cdot \\ \cdot \\ \delta_i \\ \cdot \\ \cdot \\ \delta_{N-1} \\ S \end{bmatrix}
$$

where, $A_i = \mu/D_{i-1}$, $D_i = D_1 - A_1\lambda$, $Q_i = -A_i Q_{i-1}$, $P_i = -P_{i-1}\lambda/D_i$,

$PIQI = \sum_{k=1}^{I} P_k Q_k$, then $D_i M_i + \lambda M_{i+1} + Q_i M_N = \delta_i$, $i = 1, 2, \ldots, N-1$

and

C. $\overline{D}_N M_N = S$

Common Blocks

/SPLNCO/

Variable Description

A Array A(I) = μ/D(I-1) used as multiplier in elimination procedure.

ALM $\lambda = \frac{1}{2}$

AM Moments M(I) passed in SPLNCO.

AMU $\mu = \frac{1}{2}$

C1 First coefficient of cubic.

C2 Second coefficient of cubic.

C3	Third coefficient of cubic.
C4	Fourth coefficient of cubic.
D	Array D(I) of final diagonal entries of matrix after elimination. Last diagonal element is special.
D_N	$\overline{D}_N$: diagonal element in last row after elimination.
F	Array F(I) of data around band to which spline is fitted.
FAC	$3/\Delta\lambda_j{}^2$
FLN	$hi = \frac{1}{\Delta\lambda_j}$ equal intervals.
GI	$x_i{}^2(C4x_i + C3)$
GIP	$x_{i+1}{}^2(C4x_{i+1} + C3)$
I	Index for loops.
II	Index accesses arrays from bottom to top.
IM	I - 1
IP	I + 1
N	NBJ passed by DISPLY.
NM	NBJ - 1
P	Array P(I) of multipliers in elimination.
PIQ	Array PIQ(I).
Q	Array Q(I) of entries in last column after elimination procedure.
RFLN	$\Delta\lambda_j$
S	Last component of right hand sector after elimination.
T	$\frac{1}{\Delta\lambda_j}(M_{i+1} - M_i)$
TPI	2π
XI	$\lambda_{ij} = (I - 1)\Delta\lambda_j$
XIP	$XI + \Delta\lambda_j$

Helpful Comments

SUBROUTINE SPREP(N)   (GSPLN.14)

First entry calculates the arrays A, D, Q, P, and PIQ for the elimination process. Since these arrays will remain the same whatever the right hand side, SPREP need only be called once at the beginning of DISPLY. N is not used here.

ENTRY(SPEP2)   (GSPLN.34)

This entry calculates values which appear in the elimination procedure. In particular, the value of N = NBJ is used to calculate $1/\Delta\lambda_j$, $3/\Delta\lambda_j^2$ and $\overline{D}_N$, which appear in equations A., B., and C. respectively.

DN = D(N) + Q(N) - P(NM) * ALM - PIQ(NM)   (GSPLN.40)

$$\overline{D}_N = D_N + Q_N - P_{N-1}\lambda - PIQ_{N-1}$$

ENTRY SPCOFP   (GSPLN.42)

This entry computes the coefficients of the spline, and the second derivatives.

F(N+1) = F(1)   (GSPLN.44)

$y_{n+1} = y_1$ periodic case.

AM(1) = (F(2) + F(N) - 2 * F(1)) * FAC   (GSPLN.45)

$$M_1 = \frac{3}{\Delta\lambda_j^2} (y_2 + y_N - 2y_1) = d_1$$

AM(I) = (F(IP) + F(IM) - 2. * F(I)) * FAC - AM(IM) * A(I)

Accumulates the values of the right hand vector in the AM array. That is $M_i = d_i - A_i M_{i-1}$; in particular:

$$M_N = d_N - A_N(d_{N-1} - A_{N-1}(d_{N-2} \cdot \cdot \cdot))$$

S = S + P(IM) * AM(IM)   (GSPLN.49)

Accumulates the values of the last component of the right hand vector. That is, on leaving the loop:

$\tilde{S} = P_1 d_1 + P_2(d_2 - A_2 d_1) + \ldots$; the actual value of the last component:

$$S = d_N - A_N(d_{N-1} - A_{N-1}(d_{N-2} - \ldots)) - P_1 d_1 - P_2(d_2 - A_2 d_1) \ldots$$

$$= (M_N - \tilde{S}) \text{ above.}$$

AM(N) = (AM(N) - S)/DN (GSPLN.50)

$M_N = S/\bar{D}_N$ gives actual value of M_N.

3 AM(I) = (AM(I) - ALM * AM(IP) - Q(I) * AM(N))/D(I)

Loop solves for the M_i $i = N-1, \ldots, 1$. We already have M_N.

From C. $M_i = (\delta_i - \lambda M_{i+1} - Q_i M_N)/D_i$ but we've accumulated δ_i in M_i.

X(I) = XIP (GSPLN.58)

$X_i = (i-1)\Delta\lambda_j$

4 C1(I) = F(I) - GI - C2(I) * X(I) (GSPLN.65)

Since from A.

$$S\Delta(x) = \frac{M_i}{6\Delta\lambda_j}(x_{i+1} - x)^3 + \frac{M_{i+1}}{6\Delta\lambda_j}(x - x_i)^3 + y_i - \left(\frac{M_i \Delta\lambda_j^2}{6}\right)\left(\frac{x_{i+1} - x}{\Delta\lambda_j}\right)$$

$$+ \quad y_{i+1} - \left(\frac{M_{i+1}\Delta\lambda_j^2}{6}\right)\left(\frac{x - x_i}{\Delta\lambda_j}\right)$$

then, expanding this and replacing x_{i+1} by $x_i + \Delta\lambda_j$

$$S\Delta(x) = C_4(i)x^3 + C_3(i)x^2 + C_2(i)x + C_1(i)$$

SUBROUTINE SPINT(M,D,FLD,N,J)

SPINT returns to DISPLY the array FLD, which, previous to the call, had only NBJ = N grid values in its J^{th} column. These values are the cubic spline interpolated values around the J^{th} data band, calculated so that all bands will have NBJX interpolated values on the same longitudinal slices.

Common Blocks

/SPLNCO/

Variable Description

ARG $x = \alpha_j \Delta\lambda_{j_x} + (i-1)\Delta\lambda_{j_x}$ where α_j is the fractional deviation of the j^{th} data band (indexed j+1).

D α_j passed from DISPLY.

DARG $\Delta\lambda_{j_x} = 2\pi/\text{NBJX}$, where j_x is the index of the band with maximum cells NBJX.

FLD Data field to be interpolated.

L Location of interpolated argument x in successively indexed splines, if the splines are indexed east from the zero meridian.

M NBJX

N NBJ for current band.

Helpful Comments

DARG = TPI/FLOAT(M) $ ARG = D(J+1) * DARG $ L = 1

$$\Delta\lambda_{j_x} = \text{DARG} = 2\pi/\text{NBJX}$$

The first argument is $x = \alpha_j \Delta\lambda_{j_x}$. This places x on a reduced scale within its own first displaced box. Since $\Delta\lambda_{j_x} < \Delta\lambda_j$ this x is clearly within the first indexed spline L = 1.

FLD(I,J) = ((C4(L) * ARG + C3(L)) * ARG + C2(L) * ARG + C1(L)

Displaces the i^{th} data value on band j with the interpolated data value.

ARG = ARG + DARG $ L = FLN * ARG $ L = MOD(L,N) + 1

Next argument is $x + \Delta\lambda_{j_x}$. For this x: $L = \left[\frac{x}{\Delta\lambda_j}\right]^{INT} + 1$ gives the location of the argument in the indexed splines. L = N occurs if J = Jx. Then MOD(NBJX,NBJX) + 1 = 1 is ok.

SUBROUTINE CONTOR(Z,IZ,JZ,X,Y,M,N,XT,YT,T,K,HF,HD,HH,NTAPE)

Produces contour plot of the function Z(X,Y), which is given for the grid points (X(I),Y(J)), I - 1,. . .,M and J = 1,. . .,N. The output is a sequence of strings, for the plotter, in the format (L,H) (X(1),Y(1),X(2), Y(2),. . .,X(LL,Y(LL)) where L = 2 * LL and $L \leq K$. The output is terminated

by a string having L = -1. Each string describes a contour (or part of a contour) at the level Z = H.

Common Blocks

/PLC/

Variable Description

BLANK	Option to set certain z values to 5HBLANK so that they may be skipped in processing. Not used here.
DX	x increment returned by plotting routine SCALE (which scales x array to fit over axis of 10 inches.
DY	y increment returned by plotting routine scale.
FCT	An amplification factor for characters LLINE $\geq$ 13.
H	Current contour level.
HD	Increment between contour levels.
HDD	Increment within current contour level = HD/1000
HF	First contour level = $z_{min} + z_F > z_{min}$
HH	Last contour level = $z_{max} - z_F < z_{max}$
HLEVEL	Current level or preceding level
IFLAG	IFLAG = 1 CONTINUE SEARCH FOR LINKS IFLAG = 2 OUTPUT STRING READY
II	Determines search area for neighbors in link.
IL	Zonal index of preceding link point.
ISYMB	Not used.
IS2	Index for loops.
IX	Current length of output string.
IZ	Zonal dimension of z array.
I1	Indices K1,K2,K3,K4 during search for neighboring points.
I47	I47=1 Reached mazimum size of output string I47=2 Continue processing
JJ	Determines search area for neighbors in link.

JL	Meridional index of preceding link point.

JLINE	JLINE = 1 produces a line plot with symbols at every point.

JZ	Meridional dimension of Z array.

K	Maximum size of output string.

KEY	Code passed to subroutine NEIBOR
	KEY = 1	Current point on a vertical link.
	KEY = 2	Current point on a horizontal link.

KIKI	Keeps count of number of HDD increments that are added whenever current level passes through function value.

KL	Value of KEY for preceding point.

K1, K2, K3, K4	Codes search for neighboring points.

LLINE	A number describing the centered symbol to be used 1 ≤ LLINE ≤ 13.

LSYMB	Array holds list of plotting characters used for table to right of plot.

M	Length of X table.

MM1	M - 1.

N	Length of Y table.

NLABX	Passed value is NLCX = 40 from DISPLY. Determines number of characters in X axis label.

NLABY	Passed value os NLCY = 40 from DISPLY. Determines number of characters in Y axis label.

NM1	N - 1.

NNLABX NNLABX = -NBLABX. Sign determines orientation of characters, clockwise or counterclockwise to axis.

NSTEPS Passed value NCL from DISPLY. Number of contour lines in plot.

N1, N2	Determine search area.

N3
N4 } Determine search area.

SKALE Scale factor in plotting (now 1.0).

SYSZ Height of characters plotted by SYMBOL.

T Output string temporary array.

UD $UD = |x_{max} + y_{max} + 1|$ is used to flag points already processed.

X Table of zonal grid points in degrees.

XF Zonal value of first point of a string.

XMIN Minimum X after scaling to 10 inches.

XT Temporary array of interpolated λ values for crossings at level H in the horizontal direction.

Y Table of meridional grid points in degrees.

YF Meridional value of first point of a string.

YMIN Minimum value of Y after scaling to 10 inches.

YSYMB Vertical coordinate of leftmost corner of plotted character.

YT Temporary array of interpolated θ values for crossings at level H in the vertical direction.

Z Table of functional values at grid points X, Y.

ZD $Z_{max}-Z_{min}$

ZF $Z_D/2*NCL = \Delta Z$

ZMAX Maximum value in Z table.

ZMIN Minimum value in Z table.

SUBROUTINE DPFT

Returns the 2-dimensional spectrum of the data field FLD (containing the variable specified in DISPLY at a fixed level). A restriction is that all latitudes should contain the same number of grid points (NBJX).

Common Blocks

/DPC/

Variable Description

C Temporary variable holding odd components of north-south sweep.

CZ Data array or spectral array returned by FFT.

FAC1, FAC2, FAC3, FAC4 — Normalizing factors

IA NBJX/2 + I + 1.

IX FFT code.

NT 2JB set in DISPLY.

N2 NBJX/2 set in DISPLY.

N2P N2 + 1.

S Temporary variable holding even components of N-S sweep.

SV $Z_I(2)$.

The logic of the loops is:

```
┌────── J = 1,JB
│  ┌─── I = 1,n
│  │    Z_i = F_J(i)           Fill CZ with data from band J
│  └─── 6
│       Z = FFT(Z)             Replace data in CZ with E-W spectral components
│  ┌─── I = 2,2JB
│  │    F_J(i) = 4/n   Z odd
│  │    F_J(i) = -4/n  Z even
│  └─── 7
│       F_J(n+2) = (1/n)Z_2
│       F_J(1) = (1/n)Z_1
│       F_J(n+2) = F_J(n+2) = F_J((n/2)+2) = 0
└────── 1
```

```
——I = 1,(n/2)+1
  ——J = 1,JB
    Zj = FI(j)            Fill cz with E-W components around zone i.
    Zj+JB = FI(JB1-j)     Augment data using Z(n/2)+j = Z(n/2)-j; which
  ——3                     for real data ==> Z(n/2)+j = Z(n/2)-j
    Z = FFT(Z)     Replace data in cz with N-S sweep of E-W components.
  ——J = 2,JB
    FI(j) = 4/n  Z odd
    FI(JB1+j) = -4/n  Z even
  ——4
——5  FI(JB1) = (1/n)Z2 $ FI(JB2) = 0
```

SUBROUTINE FFT(CZ,N,INV,IRE,IX)

Returns the spectrum of the complex data array CZ. The transformed data replaces the original data in CZ. If the original data is a real sequence $\{x_j\}$ $j = 0, 1, \ldots, n-1$ we must have $n = 2^k$, with $k \geq 2$. Then

$$x_j = a_o/2 + \sum_{k=1}^{(n/2)-1} \left[a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)\right]$$

$$+ (a_{n/2})/2 \cos(\pi j) \quad j = 0, 1, \ldots, n-1$$

with $$a_k = 2/n \sum_{j=0}^{n-1} k_j \cos(2\pi jk/n) \quad k = 0, 1, \ldots, n/2$$

$$b_k = 2/n \sum_{j=0}^{n-1} x_j \sin(2\pi jk/n) \quad k = 1, 2, \ldots, n/2 - 1.$$

In the more general complex case

$$z_j = \sum_{k=0}^{n-1} \alpha_x \exp\left(\pm i \frac{2\pi jk}{n}\right) \quad j = 0, 1, \ldots, n-1$$

where α_k is determined by the inverse transformation

$$\alpha_k = 1/n \sum_{j=0}^{n-1} z_j \exp\left(\mp i \frac{2\pi jk}{n}\right) \quad k = 0, 1, \ldots, n-1.$$

Then $\alpha_o = a_o/2$, $\alpha_{(n/2)} = a_{(n/2)}/2$

$$\left.\begin{aligned} \alpha_k &= \frac{a_k + ib_k}{2} \\ \alpha_{n-k} &= \frac{a_k - ib_k}{2} \end{aligned}\right\} \quad k = 1, 2, \ldots, n/2 - 1.$$

For the forward transform (INV = -1), and real data (IRE = -1)

$$\begin{bmatrix}\alpha_o\\ \alpha_1\\ \cdot\\ \cdot\\ \cdot\\ \alpha_{n-1}\end{bmatrix} \xrightarrow{\quad T \quad} \begin{bmatrix} \left.\begin{matrix}(z_o,\ z_{n/2})\\ z_1\\ \vdots\\ z_{(n/2)-1}\end{matrix}\right\} \text{Returned}\\ z_{(n/2)+1}\\ \vdots\\ z_{n-1}\end{bmatrix}$$

where $t_{jk} = \exp\left\{i\frac{2\pi jk}{n}\right\}$, and z_j for $1 \leq n \leq n-1$ is complex. Since z_o and $z_{n/2}$ are real, we need only return the first (n/2) + 1 coefficients on top of the original data, by using the following method. z_o and $z_{n/2}$ occupy the first complex number (2 words). Then z_1 thru $z_{(n/2)-1}$ occupy the next n-2 words, replacing the original real n data words, with the transformed values. Then since

$$z_{(n/2)+r} = \sum_{k=0}^{n-1} \alpha_k \exp\left(i\,\frac{2\pi(n/2+r)k}{n}\right) \qquad r = 1, 2, \ldots, n/2-1$$

$$\Longrightarrow \quad \overline{z}_{(n/2)+r} = \sum_{k=0}^{n-1} \overline{\alpha}_k \exp\left(-i\,\frac{2\pi(n/2+r)k}{n}\right)$$

$$= \sum_{k=0}^{n-1} \overline{\alpha}_k \exp\left(i\,\frac{2\pi(n/2-r)k}{n}\right)$$

$$= \sum_{k=0}^{n-1} \alpha_k \exp\left(i\,\frac{2\pi(n/2-r)k}{n}\right) = z_{(n/2)-r}$$

since for real data $\overline{\alpha}_k = \alpha_k$.

Then, returning $z_o, z_1, \ldots, z_{(n/2)-1}, z_{n/2}$

we can recover $z_{(n/2)+1}, \ldots, z_{n-1}$ from $z_{(n/2)+r} = \overline{z_{(n/2)-r}}$

For the more general complex case $\alpha_{n-s} = \overline{\alpha_s}$. For the forward transform with real data then, it is enough to sequence the even indexed data as the real part, and the odd indexed data as the imaginary part of CZ.

For the inverse transform (INV = +1) and complex data (IRE = +1)

$$\alpha_k = \frac{1}{n}\sum_{j=0}^{n-1} z_j \exp\left(-i\,\frac{2\pi jk}{n}\right) \qquad k = 0, 1, \ldots, n-1$$

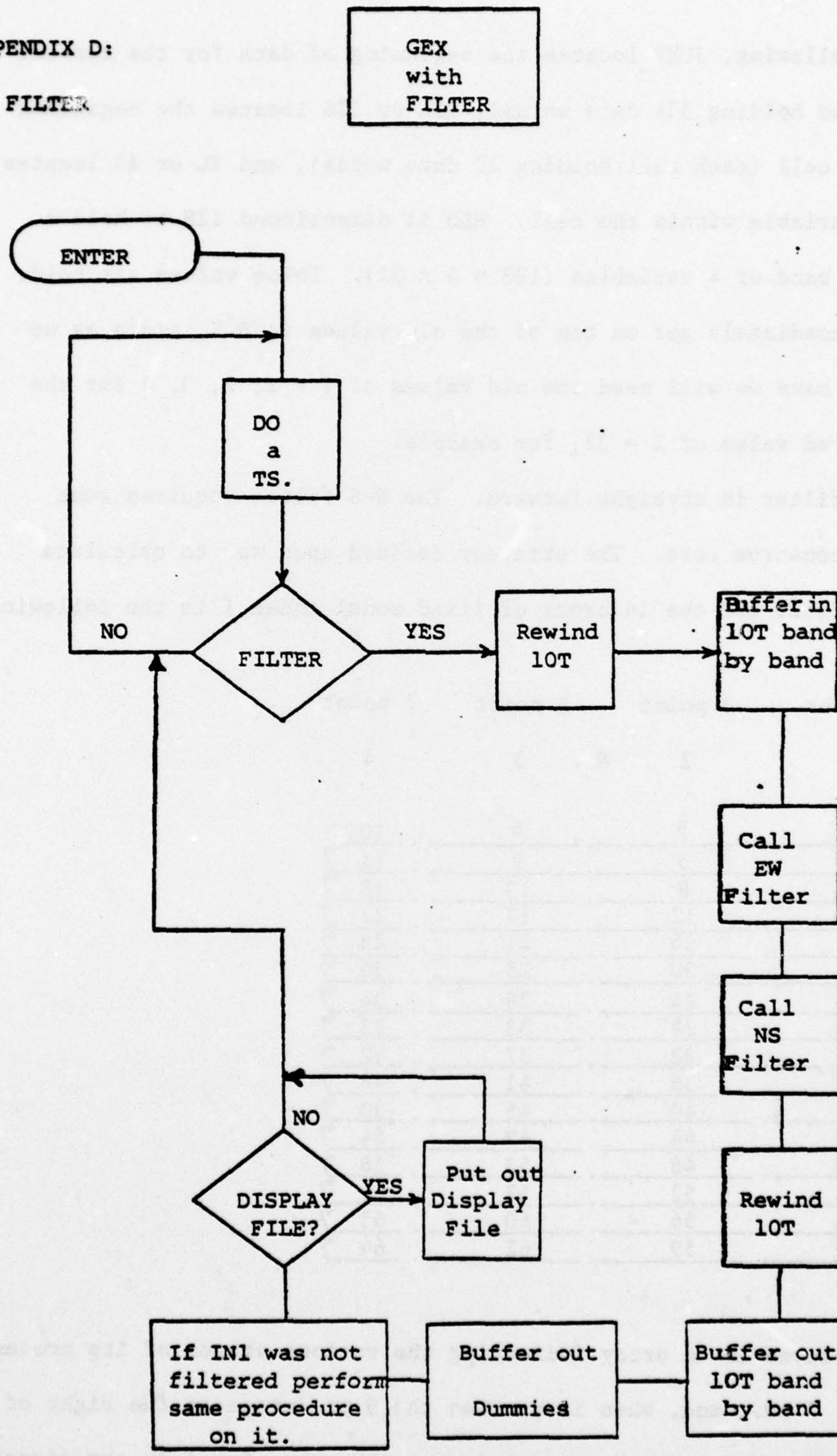
$$\begin{bmatrix} z_o \\ z_1 \\ \cdot \\ \cdot \\ \cdot \\ z_{n-1} \end{bmatrix} \xrightarrow{\quad T^* \quad} n \begin{bmatrix} \alpha_o \\ \alpha_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{n-1} \end{bmatrix}$$

so that $\vec{\alpha} = \frac{1}{n} T^* \vec{z}$.

Then the values returned by the forward transform should be rearranged and augmented using $z_{n-j} = \overline{z_j}$. The normalization by $\frac{1}{n}$ is not done in FFT itself.

APPENDIX D:

FILTER

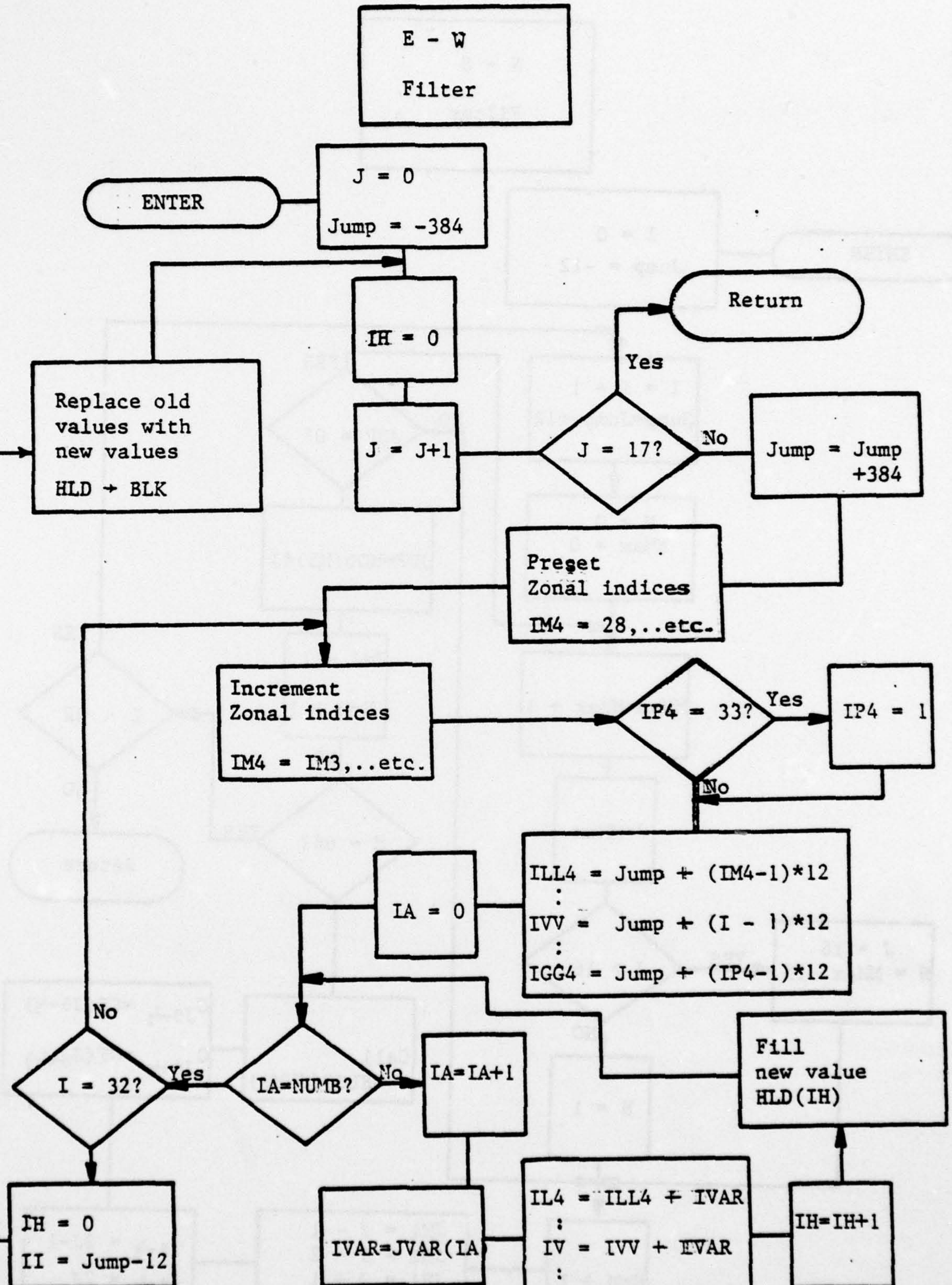


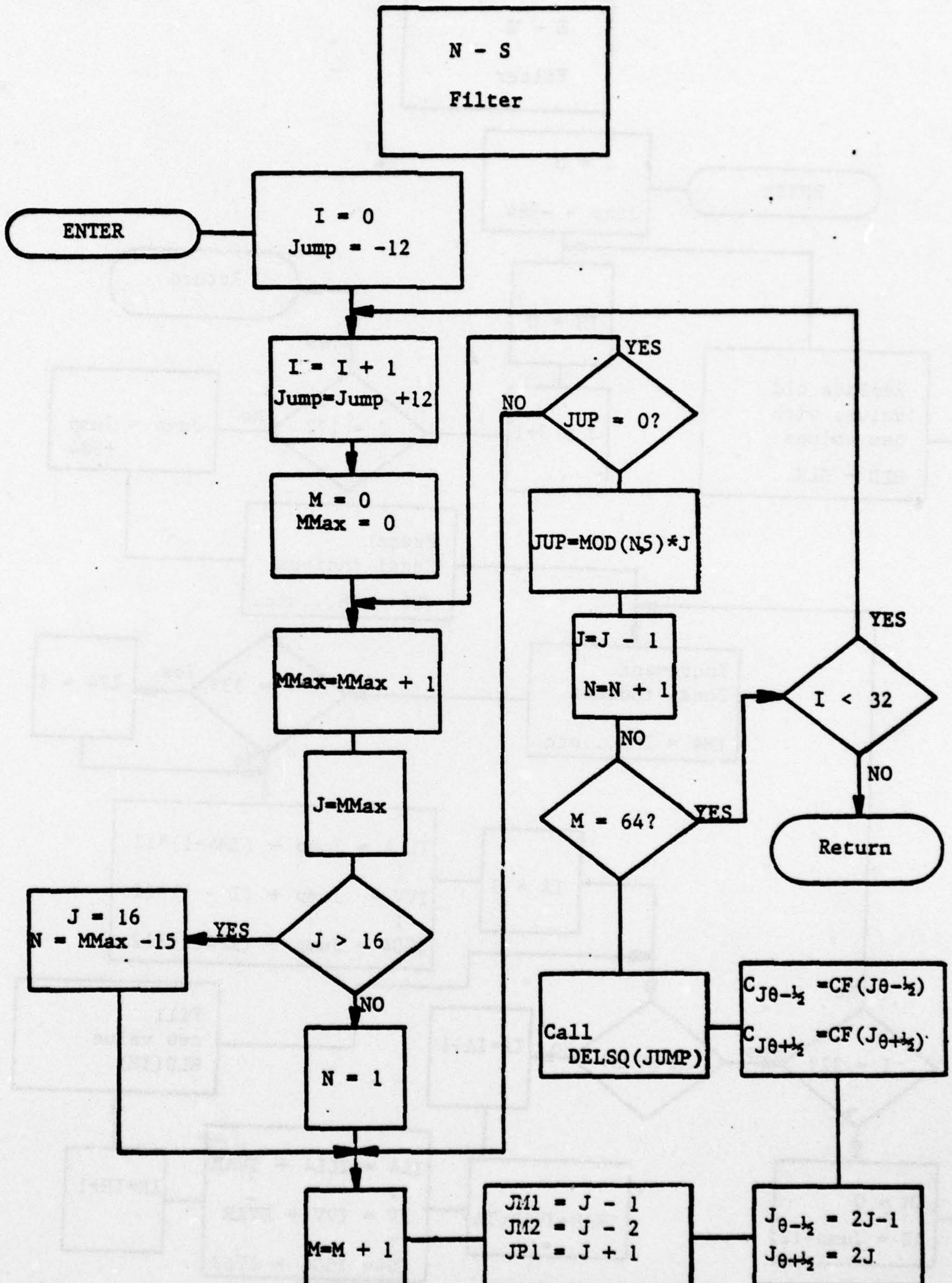
In the following, JUMP locates the beginning of data for the current band (each band holding 384 data words), ILL or IGG locates the beginning of data for a cell (each cell holding 12 data words), and IL or IG locates the current variable within the cell. HLD is dimensioned 128 to hold a full filtered band of 4 variables (128 = 4 x 32). These values are held, rather than immediately put on top of the old values in BLK, since as we go around the band we will need the old values of I = 1, 2, 3, 4 for the 9 point filtered value of I = 32, for example.

The E-W filter is straight forward. The N-S filter requires some ingenuity to conserve core. The strategy decided upon was to calculate the 9 point filter for the 16 bands of fixed zonal index I in the following pattern:

	1 point	3 point	5 point	7 point
J \ N	1	2	3	4
1	1	3	6	10
2	2	5	9	14
3	4	8	13	18
4	7	12	17	22
5	11	16	21	26
6	15	20	25	30
7	19	24	29	34
8	23	28	33	38
9	27	32	37	42
10	31	36	41	46
11	35	40	45	50
12	39	44	49	54
13	43	48	53	58
14	47	52	57	61
15	51	56	60	63
16	55	59	62	64

Data is saved in an array indicating the various stages of its processing 3 pt., 5 pt., 7 pt., and, when it reaches the 9 pt. stage on the right of the diagram, it replaces the original data on the same level in the diagram.


E - W
Filter
ENTER
J = 0
Jump = -384
IH = 0
Return
Yes
Replace old values with new values
HLD → BLK
J = J+1
J = 17?
No
Jump = Jump +384
Preset Zonal indices
IM4 = 28,..etc.
Increment Zonal indices
IM4 = IM3,..etc.
IP4 = 33?
Yes
IP4 = 1
No
IA = 0
ILL4 = Jump + (IM4-1)*12
IVV = Jump + (I - 1)*12
IGG4 = Jump + (IP4-1)*12
No
I = 32?
Yes
IA=NUMB?
No
IA=IA+1
Fill new value HLD(IH)
IH = 0
II = Jump-12
IVAR=JVAR(IA)
IL4 = ILL4 + IVAR
IV = IVV + EVAR
IG4= IGG4 + IVAR
IH=IH+1


N - S
Filter
ENTER
I = 0
Jump = -12
I = I + 1
Jump=Jump +12
M = 0
MMax = 0
MMax=MMax + 1
J=MMax
J > 16
YES
J = 16
N = MMax -15
NO
N = 1
M=M + 1
JM1 = J - 1
JM2 = J - 2
JP1 = J + 1
$J_{\theta-½}$ = 2J-1
$J_{\theta+½}$ = 2J
$C_{J\theta-½}$ =CF(Jθ-½)
$C_{J\theta+½}$ =CF($J_{\theta+½}$)
Call
DELSQ(JUMP)
M = 64?
YES
NO
J=J - 1
N=N + 1
JUP=MOD(N,5)*J
JUP = 0?
YES
NO
I < 32
YES
NO
Return

When the circled number (M) reaches 64, the process begins again with data from the next zonal index I + 1. JUMP locates cells of that zonal index within each band, with excessive data values separated by 384 words.

Then

$$\delta^2 g_j = \frac{\cos\theta_{j-\frac{1}{2}}}{\cos\theta_j} g_{j-1} - 2\cos\alpha g_j + \frac{\cos\theta_{j+\frac{1}{2}}}{\cos\theta_j} g_{j+1}$$

where the coefficients CJHM = $\frac{\cos\theta_{j-\frac{1}{2}}}{\cos\theta_j}$ and CJHP = $\frac{\cos\theta_{j+\frac{1}{2}}}{\cos\theta_j}$ have been calculated in GEX, and stored in the array CF.

DELSQ returns δ^2 of the variables in position M- successive calls to DELSQ yield:

for N = 1 $\delta^2 g_j$ where g_j is held in BLK

for N = 2 $\delta^2(\delta^2 g_j)$ where $\delta^2 g_j$ is held in the 3 pt. arrays

for N = 3 $\delta^2(\delta^4 g_j)$ where $\delta^4 g_j$ is held in the 5 pt. arrays

for N = 4 $g_{F_j} = \left[1 - \frac{\delta^2}{256}(\delta^6)\right] g_j$ where $\delta^6 g_j$ is held in the 7 pt. array, and g_{F_j} replaces the values of g_j in the array BLK.

For example, the first 10 calls to DELSQ would produce:

	1pt.	3pt.	
M=1	1.	1.	
	2.		
J=1	3.		
N=1			(JM1,3)=0

	1pt.	3pt.	
M=2	1.	1.	
	2.	2.	
J=2	3.		
N=1			(JM1,3)=1

	1pt.	3pt.	5pt.
M=3	1.	1.	1.
	2.	2.	2.
J=1	3.		
			(JM2,3)=-1
N=2			(JM1,3)=0
			(J,3)=1

	1pt.	3pt.	5pt.
M=4	1.	1.	1.
	2.	2.	
J=3	3.	3.	
N=1	4.		(JM1,3)=2

	1pt.	3pt.	5pt.
M=5	1.	1.	1.
	2.	2.	2.
J=2	3.	3.	(JM2,3)=0
N=2	4.		(JM1,3)=1
			(J,3)=2

	1pt.	3pt.	5pt.	7pt.
M=6	1.	1.	1.	1.
	2.	2.	2.	
J=1	3.	3.	(JM2,3)=-1	
N=3	4.		(JM1,3)=0	
			(J,3)=1	

	1pt.	3pt.	5pt.	7pt.
M=7	1.	4.	1.	1.
	2.	2.	2.	
J=4	3.	3.		
	4.			
N=1	5.		(JM1,3)=0	

	1pt.	3pt.	5pt.	7pt.
M=8	1.	4.	1.	1.
	2.	2.	2.	
J=3	3.	3.	3.	
	4.		(JM2,3)=1	
N=2	5.		(JM1,3)=2	
			(J,3)=0	

	1pt.	3pt.	5pt.	7pt.
M=9	1.	4.	1.	1.
	2.	2.	2.	2.
J=2	3.	3.	3.	
	4.		(JM2,3)=0	
N=3	5.		(JM1,3)=1	
			(J,3)=2	

	1pt.	3pt.	5pt.	7pt.
M=10	1.	4.	1.	1.
	2.	2.	2.	2.
J=1	3.	3.	3.	
	4.		(JM2,3)=-1	
N=4	5.		(JM1,3)=0	
			(J,3)=1	

Values out of array bounds, as for example BLK(JSP1) for J = 16, are multiplied by the coefficient zero!

```
LEON,CM110000,T3..TWO STEP RUN OF FILTER.          2317    LEONARD
ATTACH,OLDPL,GGX2974,ID=METTAUER,MR=1.
ATTACH,TAPE7,IN,REGGX2974,ID=METTAUER,MR=1.
ATTACH,TAPE3,SRLREGGX2974,ID=METTAUER,MR=1.
REQUEST,TAPE9,*PF.
UPDATE(P,Q.
FTN,OPT=2,I=COMPILE,SL.
FTN,OPT=2,SL.
PAUSE. OPERATOR STOP ON SSW41
LDSET(PRESET=ZERO)
LGO.
CATALOG,TAPE9,SLGG.TIMESX2974,ID=METTAUER,MR=1,RP=999.
*IDENT TSET
*D GGEX.3
      1TAPE8=0,TAPE9=0,TAPE10=0,TAPE11=0,TAPE12=0,TAPE13=0)
*I GGEX.6
      COMMON/COEF/CF(32)
      COMMON/FILT/JVAR(4),NUMB
      COMMON/OV256/OV256,CJ
      DIMENSION IARRAY(560)
      DIMENSION CLK(3)
      EQUIVALENCE (CLK(1),NSAVE),(CLK(2),MSAVE)
*I GGEX.11
      DATA IDES2/13/
*I GGEX.16
      REWIND IDES2
*I GGEX.25
      READ(INPUT,150) IFX,IFIL,NUMB,(JVAR(I),I=1,NUMB)
  150 FORMAT(3I5/4I5)
      WRITE(IPRT,151) IFX,IFIL,NUMB,(JVAR(I),I=1,NUMB)
  151 FORMAT(1H0,50X,*FILTER PARAMETERS*///* CODE FOR LAST TIME STEP IFX
     1=*4X,I5/* FREQUENCY OF FILTER IFIL=*5X,I5/* NUMBER OF VARIABLES TO
     2 FILTER NUMB=*I5/* CODE FOR VARIABLES*/* U=1*/* V=2*/* T=3*/* R=4*
     3/* PST=5*//4I5)
      ALPHA=.5*(BAND(2)-BAND(3))
      DO 160 J=2,17
      JO=2*J-3 $ JE=JO+1
      JP1=J+1
      BANDJ=BAND(J)$BANJP1=BAND(JP1)
      CF(JO)=COS(BANDJ)/CTHJ(J)$CF(JE)=COS(BANJP1)/CTHJ(J)
  160 CONTINUE
      CJ=-2.*COS(ALPHA)
      CF(1)=CF(32)=0.
      OV256=1./256.
*I GGEX.29
      NKV=K/2$KVM=KV-1
*I GGEX.30
      NDES2=0
      NF=16161616161616161616B
      IF=06060606060606060606B
      MSK=77000000000000000000B
      N=0
```

BEST AVAILABLE COPY

```
*I GGEX.39
      N1FX=1
      GO TO 140
*I GGEX.42
      N1FX=IFX
  140 CONTINUE
      IF(MOD(N1S,IFIL).EQ.0) GO TO 37
      M=(N+10)/10
      ISHFT=60+(-6)*MOD(N,10)
      I1=MXIFIX(MSK,ISHFT)
      I2=I1.AND.NF
      IAPRAY(M)=IARRAY(M)+I2
      N=N+1
      IFX=0$GO TO 39
   37 CONTINUE
      IFX=1
      M=(N+10)/10
      ISHFT=60+(-6)*MOD(N,10)
      I1=MXIFIX(MSK,ISHFT)
      I2=I1.AND.IF
      IARRAY(M)=IARRAY(M)+I2
      N=N+1
      REWIND IOT
      UN=UNIT(IOT)$BUFFER IN(IOT,1) (CLK(1),CLK(3))
      DO 500 J=1,16
      UN=UNIT(IOT)
      J2=384*J
      J1=J2-383
      BUFFER IN(IOT,1) (BLK(J1),BLK(J2))
  500 CONTINUE
      UN=UNIT(IOT)
      CALL FILTEW
      CALL FILTNS
      REWIND IOT
      UN=UNIT(IOT)$BUFFER OUT(IOT,1) (CLK(1),CLK(3))
      DO 501 J=1,16
      J2=384*J
      J1=J2-383
      UN=UNIT(IOT)
      BUFFER OUT(IOT,1) (BLK(J1),BLK(J2))
  501 CONTINUE
      UN=UNIT(IOT)
      DO 3 I=1,3
    3 CALL RBUFT(IOT,L,999,U,BLK,3)$CALL CBUFT(IOT,4,999,U,BLK,3)
      REWIND IOT
      IF(N1FX.EQ.1) GO TO 36
      REWIND IN1
      UN=UNIT(IN1)$BUFFER IN (IN1,1) (CLK(1),CLK(3))
      DO 506 J=1,16
      J2=384*J
      J1=J2-383
      UN=UNIT(IN1)
      BUFFER IN (IN1,1) (BLK(J1),BLK(J2))
```

```
555 CONTINUE
    UN=UNIT(IN1)
    CALL FILTEW
    CALL FILTNS
    REWIND IN1
    UN=UNIT(IN1)$BUFFER OUT(IN1,1) (CLK(1),CLK(3))
    DO 502 J=1,16
    J2=384*J
    J1=J2-383
    UN=UNIT(IN1)
    BUFFER OUT(IN1,1) (BLK(J1),BLK(J2))
502 CONTINUE
    UN=UNIT(IN1)
    DO 4 I=1,3
  4 CALL PBUFT(IN1,L,910,U,BLK,3)$CALL CBUFT(IN1,4,910,U,BLK,3)
    REWIND IN1
 36 CONTINUE
    IF(MOD(NTS,IDESF).NE.0) GO TO 39
    IF(NTS.NE.MTS) GO TO 40
    UN=UNIT(IN1)
    CALL COMP(IN1,IDES2)
    ENDFILE IDES2
    NDES2=NDES2+1
    REWIND IN1
 40 CONTINUE
    UN=UNIT(IOT)
    CALL COMP(IOT,IDES2)
    ENDFILE IDES2
    NDES2=NDES2+1
    REWIND IOT
*D GGEX.43,GGEX.45
 39 IF(NTS.LT.MTS) GO TO 34
    WRITE(IPRT,108) (IARRAY(K),K=1,M)
108 FORMAT(1H0,12A10)
    WRITE(IPRT,1010) NDES,NDES2,NLEX,IEX$REWIND IDES$REWIND IDES2$STOP
1010 FORMAT(*0TOTAL DISPLAY FILES GENERATED *,2I5/*0 CODES *2I5)
*COMPILE GGEX,GTTSLF,GSTTSLF
*COMPILE GFRC1,GTP,GFSY1,GPHIPP,GPSPL,GPST1,GYSL1
*COMPILE GROTATE,GPHS,GENTIS,GOE,GBUF
*COMPILE GSTISE,GTISE
    SUBROUTINE FILTEW
    COMMON/BUF/BLK(6528)
    COMMON/FILT/ JVAR(4),NUMP
    COMMON/CV256/CV256,CJ
    DIMENSION HLD(128)
    J=0$JUMP=-384
  1 CONTINUE
    IH=0
    J=J+1
    IF(J.EQ.17) GO TO 5
    JUMP=JUMP+384
    IM4=28$IM3=29$IM2=30$IM1=31 I=32$IP1=1$IP2=2$IP3=3$IP4=4
```


BEST AVAILABLE COPY

```
  2 CONTINUE
    IM4=IM3$IM3=IM2$IM2=IM1$IM1=I$I=IP1$IP1=IP2$IP2=IP3$IP3=IP4
    IP4=IP4+1
    IF(IP4.EQ.33) IP4=1
    ILL4=JUMP+(IM4-1)*12$ILL3=JUMP+(IM3-1)*12
    ILL2=JUMP+(IM2-1)*12$ILL1=JUMP+(IM1-1)*12$IVV=JUMP+(I-1)*12
    IGG1=JUMP+(IP1-1)*12$IGG2=JUMP+(IP2-1)*12$IGG3=JUMP+(IP3-1)*12
    IGG4=JUMP+(IP4-1)*12
    IA=0
  3 CONTINUE
    IF(IA.EQ.NUMB) GO TO 4
    IA=IA+1
    IVAR=JVAR(IA)
    IL4=ILL4+IVAR$IL3=ILL3+IVAR$IL2=ILL2+IVAR$IL1=ILL1+IVAR
    IV=IVV+IVAR$IG1=IGG1+IVAR$IG2=IGG2+IVAR$IG3=IGG3+IVAR
    IG4=IGG4+IVAR
    IH=IH+1
    HLD(IH)=OV256*(186.*BLK(IV)-(BLK(IL4)+BLK(IG4))+8.*(BLK(IL3)+
   1BLK(IG3))-28.*(BLK(IL2)+BLK(IG2))+56.*(BLK(IL1)+BLK(IG1)))
    GO TO 3
  4 IF(I.EQ.32) GO TO 6
    GO TO 2
  5 CONTINUE
    RETURN
  6 CONTINUE
    IH=0
    II=JUMP-12
    DO 7 I=1,32
    II=II+12
    IA=0
  8 CONTINUE
    IF(IA.EQ.NUMB) GO TO 7
    IA=IA+1
    IVAR=JVAR(IA)
    LL=II+IVAR
    IH=IH+1
    BLK(LL)=HLD(IH)
    GO TO 8
  7 CONTINUE
    GO TO 1
    END
    SUBROUTINE FILTNS
    COMMON/PSQ/CJHM,CJHP,J,N,JM1,JM2,JP1
    COMMON/BUF/BLK(6528)
    COMMON/ROUTE/U3PT(3),U5PT(3),U7PT(3),V3PT(3),V5PT(3),V7PT(3),
   1T3PT(3),T5PT(3),T7PT(3),R3PT(3),R5PT(3),R7PT(3),PST3PT(3),PST5PT(3
   2),PST7PT(3)
    COMMON/COEF/CF(32)
    COMMON/FILT/JVAR(4),NUMB
    COMMON/OV256/OV256,CJ
    I=0$JUMP=-12
```


BEST AVAILABLE COPY

```
    1 CONTINUE
      I=I+1
      JUMP=JUMP+12
      M=0$MMAX=0
    2 CONTINUE
      MMAX=MMAX+1
      J=MMAX
      IF(J.GT.16) GO TO 5
      N=1
    3 CONTINUE
      M=M+1
      JM1=J-1$JM2=J-2$JP1=J+1
      JTHM=2*J-1$JTHP=JTHM+1
      CJHM=CF(JTHM)$CJHP=CF(JTHP)
      CALL DELSQ(JUMP)
      IF(M.EQ.64) GO TO 4
      J=J-1
      N=N+1
      JJP=MOD(N,5)*J
      IF(JJP.EQ.0) GO TO 2
      GO TO 3
    4 IF(I.LT.32) GO TO 1
      RETURN
    5 CONTINUE
      J=16
      N=MMAX-15
      GO TO 3
      END
      SUBROUTINE DELSQ(JUMP)
      COMMON/DSQ/CJHM,CJHP,J,N,JM1,JM2,JP1
      COMMON/BUF/BLK(6528)
      COMMON/ROUTE/U3PT(3),U5PT(3),U7PT(3),V3PT(3),V5PT(3),V7PT(3),
     1T3PT(3),T5PT(3),T7PT(3),R3PT(3),R5PT(3),R7PT(3),PST3PT(3),PST5PT(3
     2),PST7PT(3)
      COMMON/FILT/JVAR(4),NUMB
      COMMON/CV256/CV256,CJ
      DIMENSION ARRAY(45)
      EQUIVALENCE (ARRAY(1),U3PT(1))
      IA=0
      IF((N-1)*(N-4)) 2,10,2
    2 CONTINUE
      MJV=MOD(J,3)
      MJV1=MOD(JM1,3)
      MJV2=MOD(JM2,3)
      NM2TT=3*(N-2)
      IF(IA.EQ.NUMB) RETURN
      IA=IA+1
      IVAR=JVAR(IA)
      INC=9*(IVAR-1)+1
      LOCVAR=INC+NM2TT
      JS=LOCVAR+MOV1
      JR=JS+3
```

BEST AVAILABLE COPY

```
      JSP1=LOCVAR+MOV
      JSM1=LOCVAR+MOV2
      ARRAY(JR)=CJHM*ARRAY(JSM1)+CJ*ARRAY(JS)+CJHP*ARRAY(JSP1)
      GO TO 2
   10 CONTINUE
      MOVB=384*JM1+JUMP
      MOV1=MOD(JM1,3)
      IF(N.EQ.4) GO TO 20
    3 CONTINUE
      IF(IA.EQ.NUMB) RETURN
      IA=IA+1
      IVAR=JVAR(IA)
      INC=9*(IVAR-1)+1
      JS=IVAR+MOVB
      JSM1=JS-384$IF(J.EQ.1) JSM1=JS
      JSP1=JS+384
      LOCR=INC
      JR=LOCR+MOV1
      ARRAY(JR)=CJHM*BLK(JSM1)+CJ*BLK(JS)+CJHP*BLK(JSP1)
      GO TO 3
   20 CONTINUE
      MOV=MOD(J,3)
      MOV2=MOD(JM2,3)
      IF(IA.EQ.NUMB) RETURN
      IA=IA+1
      IVAR=JVAR(IA)
      INC=9*(IVAR-1)+1
      JR=IVAR+MOVB
      LOCVAR=INC+6
      JS=LOCVAR+MOV1
      JSP1=LOCVAR+MOV
      JSM1=LOCVAR+MOV2
      BLK(JR)=BLK(JR)-OV256*(CJHM*ARRAY(JSM1)+CJ*ARRAY(JS)+CJHP*
     1ARRAY(JSP1))
      GO TO 20
      END
  TR=4 LF TEST FILTER
$PARAM FTI=360.,MTS=2,TRESF=1,TTT=720.,RTL=40.$
   0     1     3
   1     2     5
```

BEST AVAILABLE COPY

APPENDIX E: BERROR CODES (ICE,M)

For the codes below $1 \leq N \leq NBLK$

$2 \leq J \leq JB1$

$1 \leq L \leq 4$

SUBROUTINE	CODES
INIT	(15,1) (15,2) (16,0) (16,1)
OBAL	(20,N)
REPORT	(30,1) (31,J) (32,J) (33,J) (34,0)
STISφ	(95,0) (±60,0) (±61,0) (±62,0) (±63,0) (±64,0)
TISφ	(95,1) (±55,5) (±56,J)
GEX	(60,0) (60,1)
COMP	(±1,0) (±1,1) (±3,J) (±4,N)
ENTRY EXPAND	(±2,1) (±5,N) (±6,J) (±7,L)
ENTIS	(99,0) (99,L)
BDUMP	(40,0) (40,±1) (40,2) (40,$\overline{3}$)
DISPLAY	(0,0) (0,N)
SKFL	(1,0) (1,N)